

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ  
ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Υποβιβασμός Τάξης Μοντέλου μεγάλης κλίμακας κυκλωμάτων  
με τεχνικές Δυναμικών Συστημάτων

Model Order Reduction of large scale circuits with  
Dynamical System techniques

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Της

Παναγιωτάρα Ε. Παρασκευής

**Υπεύθυνος Καθηγητής:** Νέστωρ Ευμορφόπουλος,  
Επίκουρος Καθηγητής

**Συνεπιβλέπων Καθηγητής:** Γεώργιος Σταμούλης,  
Καθηγητής

Βόλος, Ιανουάριος 2017

Η σελίδα αυτή είναι σκόπιμα λευκή.



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ  
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Υποβιβασμός Τάξης Μοντέλου μεγάλης κλίμακας κυκλωμάτων  
με τεχνικές Δυναμικών Συστημάτων

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Της

Παναγιωτάρα Ε. Παρασκευής

**Επιβλέπων Καθηγητής:** Νέστωρ Ευμορφόπουλος,  
Επίκουρος Καθηγητής

**Συνεπιβλέπων Καθηγητής:** Γεώργιος Σταμούλης,  
Καθηγητής

*Εγκρίθηκε από την διμελή εξεταστική επιτροπή την 31<sup>η</sup> Ιανουαρίου 2017.*

.....  
Νέστωρ Ευμορφόπουλος,  
Επίκουρος Καθηγητής

.....  
Σταμούλης Γεώργιος,  
Καθηγητής

Η σελίδα αυτή είναι σκόπιμα λευκή.

Διπλωματική Εργασία για την απόκτηση του Διπλώματος του Μηχανικού Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας, στα πλαίσια του Προγράμματος Προπτυχιακών Σπουδών του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας.

.....

Παναγιωτάρα Ε. Παρασκευή

Διπλωματούχος Μηχανικός Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών Πανεπιστημίου Θεσσαλίας

Copyright © Panagiotara E. Paraskevi, 2017

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς την συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν την συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας.

Η σελίδα αυτή είναι σκόπιμα λευκή.

*Στους γονείς μου, Ευάγγελο και Ελένη  
και στην πολυαγαπημένη μου αδελφή,  
Βάλια...*

Η σελίδα αυτή είναι σκόπιμα λευκή.



## Ευχαριστήριο Σημείωμα

Με την περάτωση της παρούσης διπλωματικής εργασίας ολοκληρώνεται ένας κύκλος πέντε χρόνων προπτυχιακών σπουδών στο τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας. Η διπλωματική αυτή προπτυχιακή εργασία εκπονήθηκε κατά το ακαδημαϊκό έτος 2016-2017 και στο σημείο αυτό θα ήθελα να ευχαριστήσω όλους όσους έβαλαν ένα λιθαράκι βοηθώντας με άμεσα ή έμμεσα για να ολοκληρωθεί. Σίγουρα, αυτές οι λίγες γραμμές που ακολουθούν δεν φτάνουν για να εκφράσω την ευγνωμοσύνη μου προς όλους αυτούς όπου αυτά τα χρόνια ήταν δίπλα μου σε κάθε βήμα και με στήριζαν ο καθένας με τον δικό του, ιδιαίτερο τρόπο.

Η τρέχουσα διπλωματική εργασία εκπονήθηκε υπό την επίβλεψη του Επίκουρου Καθηγητή του τμήματος Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας κ. Νέστορα Ευμορφόπουλου στον οποίο θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα. Θα ήθελα επιπλέον να τον ευχαριστήσω για την άριστη συνεργασία που διατηρήσαμε κατά την εκπόνηση της εργασίας και για τις κατευθυντήριες γραμμές που μου παρείχε όλους αυτούς τους μήνες οποτεδήποτε τις χρειαζόμουν και οι οποίες με βοήθησαν στην έγκαιρη ολοκλήρωση αυτής της μελέτης. Ο κύριος Ευμορφόπουλος εκτός από μια αστείρευτη πηγή γνώσης και λαμπρό επιστήμονα αποτελεί και έναν πολύ προσιτό άνθρωπο που φρόντιζε καθ' όλη την διάρκεια της δύσκολης περιόδου της διπλωματικής μου εργασίας να με ενθαρρύνει και να με ωθεί στο να πιστεύω ολοένα και περισσότερο στις ικανότητες και στις δυνάμεις μου. Φυσικά, θα ήμουν αγνώμων αν παρέλειπα να ευχαριστήσω τον έτερο επιβλέποντα καθηγητή της παρούσης διπλωματικής εργασίας και Καθηγητή του τμήματος Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας κ. Γεώργιο Σταμούλη για τις συμβουλές και τις επιστημονικές του. Συγχρόνως, θα ήθελα να ευχαριστήσω όλους τους καθηγητές του τμήματος, οι οποίοι όλα αυτά τα χρόνια προσπάθησαν να μου μεταλαμπαδεύσουν αρχές, αξίες και πολύτιμα εφόδια που θα μου φανούν απολύτως απαραίτητα για ένα λαμπρό μέλλον ως Ηλεκτρολόγος Μηχανικός.

Επιπροσθέτως, θα ήθελα να ευχαριστήσω ιδιαίτερω τον διδακτορικό φοιτητή του τμήματος Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών κ. Γεώργιο Φλώρο για την πολύτιμη βοήθεια και το υλικό του, την άμεση μετάδοση γνώσεων καθώς και τις υποδείξεις που μου παρείχε σε θέματα υλοποίησης, συνδράμοντας κατά αυτόν τον τρόπο στην επιτυχή ολοκλήρωση της εργασίας μου. Ακόμη, οφείλω να ευχαριστήσω τον κ. Τσιόβουλο Απόστολο για την καθοδήγηση του σε προβλήματα τεχνικής φύσεως.

Το μεγαλύτερο ευχαριστώ όμως ανήκει στην οικογένεια μου, στους γονείς μου Ευάγγελο και Ελένη και στην αδερφή μου Βάλια, γιατί χωρίς την ηθική υποστήριξη, την απεριόριστη αγάπη και την αμέριστη κατανόηση και υπομονή τους τόσο κατά την διάρκεια των σπουδών μου όσο και κατά την διάρκεια αυτής της ερευνητικής μου μελέτης ίσως δεν είχα καταφέρει σχεδόν τίποτα. Χάριν στους δικούς τους αγώνες και θυσίες όλα αυτά τα χρόνια της ακαδημαϊκής μου πορείας κατάφερα να πραγματοποιήσω τα

περισσότερα από τα όνειρα και τους στόχους μου, τα οποία σε διαφορετική περίπτωση θα έμοιαζαν ουτοπικά και προϊόντα φαντασίας.

Τέλος, θα ήθελα να ευχαριστήσω τους συμφοιτητές μου και μελλοντικούς συναδέλφους μηχανικούς που ήταν δίπλα μου σε όλες τις δύσκολες στιγμές, γεμάτοι με αισιοδοξία και καλή διάθεση καθώς και για τις ξεχωριστές στιγμές και τα όμορφα φοιτητικά χρόνια που περάσαμε μαζί. Η θετική ενέργειά τους πάντα με βοηθούσε να ξεπερνά τα άγχη και τις ανησυχίες που καλούμουν να αντιμετωπίσω στο δύσκολο και απαιτητικό αντικείμενο του Ηλεκτρολόγου Μηχανικού.

**Παναγιωτάρα Ε. Παρασκευή**  
**Βόλος, 2017**

# Πίνακας περιεχομένων

Ευχαριστήριο Σημείωμα .....	viii
Κατάλογος Αλγορίθμων .....	xiii
Κατάλογος Εικόνων .....	xiv
Κατάλογος Σχημάτων .....	xv
Κατάλογος Πινάκων .....	xvi
Κατάλογος Συντομογραφιών .....	xvii
ΠΕΡΙΛΗΨΗ .....	xviii
ABSTRACT .....	xix
ΚΕΦΑΛΑΙΟ 1 .....	1
Εισαγωγή .....	1
1.1 Προσομοίωση σήμερα .....	1
1.2 Περιγραφή του Προβλήματος και Συμβολή της Διπλωματικής Εργασίας .....	4
1.3 Ιστορική Αναδρομή .....	6
1.4 Αποτελέσματα και Σκοπός της Διπλωματικής Εργασίας .....	6
1.5 Διάρθρωση της Διπλωματικής Εργασίας .....	7
ΚΕΦΑΛΑΙΟ 2 .....	10
Μαθηματικό Υπόβαθρο .....	10
2.1 Κατηγορίες Συστημάτων .....	10
2.2 Χώρος Κατάστασης .....	12
2.2.1 Ορισμοί Δυναμικών Συστημάτων .....	12
2.2.2 Εύρεση Συνάρτησης Μεταφοράς από Δυναμικές Εξισώσεις .....	14
2.2.3 Ευστάθεια, Παθητικότητα ΓΧΑ Συστημάτων .....	15
2.2.4 Ισοδύναμα Συστήματα .....	16

2.2.5	Μηδενικά και Πόλοι .....	17
2.3	Ελεγχιμότητα και Παρατηρησιμότητα .....	18
2.3.1	Ορισμός Ελεγχιμότητα και Παρατηρησιμότητας .....	18
2.3.2	Ταυτόχρονη Ελεγχιμότητα και Παρατηρησιμότητα .....	18
2.4	Modified Nodal Analysis (MNA) Εξισώσεις .....	19
2.5	Συσχετισμός των MNA εξισώσεων με τις Δυναμικές Εξισώσεις .....	22
2.5.1	Τερματικά και Θύρες .....	24
2.6	Έννοιες Γραμμικής Άλγεβρας .....	27
<b>ΚΕΦΑΛΑΙΟ 3 .....</b>		<b>30</b>
<b>Μέθοδοι Υποβιβασμού Τάξης Μοντέλου .....</b>		<b>30</b>
3.1	Επισκόπηση μεθόδων MOR .....	30
3.2	Ιδιαίτερα Χαρακτηριστικά της μεθόδου TBR .....	32
3.3	Φυσική Ερμηνεία του TBR Αλγορίθμου .....	35
<b>ΚΕΦΑΛΑΙΟ 4 .....</b>		<b>38</b>
<b>Ανάπτυξη Λογισμικού MATLAB .....</b>		<b>38</b>
4.1	Περιβάλλον Ανάπτυξης Αλγορίθμου .....	38
4.2	Το περιβάλλον της MATLAB .....	41
4.3	Control System Toolbox .....	43
<b>ΚΕΦΑΛΑΙΟ 5 .....</b>		<b>45</b>
<b>Υλοποίηση και βήματα αλγορίθμου TBR .....</b>		<b>45</b>
5.1	Η μέθοδος υποβιβασμού τάξης μοντέλου TBR .....	45
5.2	Υλοποίηση TBR .....	47
5.2.1	Balanced Realization .....	47
5.2.2	Διαδικασία Μείωσης Τάξης Μοντέλου .....	51
5.3	Παραλλαγή του TBR .....	53
<b>Κεφάλαιο 6 .....</b>		<b>56</b>
<b>Υλοποίηση της Προσομοίωσης .....</b>		<b>56</b>
6.1	Αρχείο περιγραφής κυκλώματος .....	56

6.2	Κατασκευή Προσομοιωτή.....	58
6.3	MNA σύστημα του κυκλώματος.....	60
6.4	Χρήση μεθόδων υποβιβασμού τάξης μοντέλου.....	61
<b>Κεφάλαιο 7.....</b>		<b>63</b>
<b>Εκτίμηση της Λύσης - Αποτελέσματα – Συγκρίσεις .....</b>		<b>63</b>
7.1	Πειραματικά Κυκλώματα.....	63
7.2	Πειραματική Διαδικασία .....	64
7.2.1	Ανάλυση υλοποίησης συναρτήσεων διπλωματικής εργασίας .....	64
7.2.2	Ενδεικτικά Αποτελέσματα Προσομοίωσης .....	70
7.3	Αποτελέσματα μελέτης .....	74
<b>Κεφάλαιο 8.....</b>		<b>76</b>
<b>Επίλογος.....</b>		<b>76</b>
8.1	Σύνοψη διπλωματικής εργασίας και των αποτελεσμάτων της.....	76
8.2	Μελλοντικές επεκτάσεις .....	77
<b>Παραρτήματα .....</b>		<b>80</b>
<b>MATLAB Κώδικες.....</b>		<b>80</b>
A	Παράρτημα Α.....	80
<b>Βιβλιογραφία .....</b>		<b>99</b>

## **Κατάλογος Αλγορίθμων**

Truncated Balanced Realization

Krylov Subspaces

Arnoldi

Proper Orthogonal Decomposition

Lanczos

## Κατάλογος Εικόνων

Εικόνα 1: Διαδικασία σχεδίασης ενός φυσικού συστήματος.....	2
Εικόνα 2: Προσομοίωση - συνδυαστικός κρίκος θεωρίας και πειράματος ...	3
Εικόνα 3: Ενδεικτική Μορφή Model Order Reduction.....	5
Εικόνα 4: Σύγχρονη μορφή ολοκληρωμένων κυκλωμάτων.....	6
Εικόνα 5: Παράδειγμα εισόδου και εξόδου ΓΧΑ συστήματος:.....	11
Εικόνα 6: Παράδειγμα ενδεικτικού κυκλώματος RLC .....	12
Εικόνα 7: Συνάρτηση Μεταφοράς.....	14
Εικόνα 8: Αναπαράσταση Μ/Σ Laplace στο μιγαδικό επίπεδο.....	17
Εικόνα 9: Σφραγίδες αντιστατών πρώτης ομάδας.....	20
Εικόνα 10: Σφραγίδες αντιστατών δεύτερης ομάδας .....	20
Εικόνα 11: Σφραγίδες πηγών ρεύματος πρώτης ομάδας.....	21
Εικόνα 12: Σφραγίδες πηγών ρεύματος δεύτερης ομάδας .....	21
Εικόνα 13: Σφραγίδες πηγών τάσης .....	21
Εικόνα 14: Κύκλωμα με τερματικό $a$ , εσωτερικό κόμβο 1, θύρα P και θύρα $Q(a,0)$ .....	24
Εικόνα 15: Κύκλωμα με N θύρες .....	25
Εικόνα 16: Κύκλωμα για εξαγωγή MNA πινάκων σε Y μορφή .....	26
Εικόνα 17: Mashup λέξεων με τα χαρακτηριστικά του TBR.....	35
Εικόνα 18: Περιβάλλον επιφάνειας εργασίας του MATLAB.....	42
Εικόνα 19: Demo χρήσης Toolbox.....	43
Εικόνα 20: Script μοντελοποίησης .....	43
Εικόνα 21: Παράδειγμα τηλεφωνικού καταλόγου σε μορφή hash table.....	59
Εικόνα 22: MatrixMarket Format αρχείου .....	59
Εικόνα 23: Απεικόνιση πίνακα C .....	65
Εικόνα 24: Μη μηδενικές εγγραφές πίνακα C .....	66
Εικόνα 25: Μη μηδενικές εγγραφές πίνακα G .....	66
Εικόνα 26: Pop-up παράθυρο μηνύματος για ευστάθεια συστήματος.....	67

Εικόνα 27: Παράθυρο εισαγωγής νέας τάξης μοντέλου συστήματος.....	67
Εικόνα 28: Μη έγκυρη τάξη μοντέλου συστήματος .....	67
Εικόνα 29: Υπενθύμιση θετικής τιμής του Reduction Error .....	68
Εικόνα 30: Επιτυχής ολοκλήρωση της μεθόδου TBR .....	69
Εικόνα 31: Επανάληψη διαδικασίας.....	69
Εικόνα 32: Ολοκλήρωση διαδικασίας.....	69
Εικόνα 33: Απόκριση συχνότητας αρχικού - τελικού μοντέλου συστήματος για την 1η είσοδο.....	71
Εικόνα 34: Απόκριση συχνότητας αρχικού - τελικού μοντέλου συστήματος για την 2η είσοδο.....	71
Εικόνα 35: Απόκριση συχνότητας αρχικού - τελικού μοντέλου συστήματος για την 3η είσοδο.....	72
Εικόνα 36: Απόκριση συχνότητας αρχικού - τελικού μοντέλου συστήματος για την 4η είσοδο.....	72
Εικόνα 37: Χρήση CPU (σε sec) παραγωγής αρχικού μοντέλου συστήματος.....	73
Εικόνα 38: Χρήση CPU (σε sec) παραγωγής ελαχιστοποιημένου μοντέλου συστήματος.....	74
Εικόνα 39: Χρήση CPU (σε sec) παραγωγής μειωμένου μοντέλου συστήματος μέσω της TBR .....	74

## Κατάλογος Σχημάτων

Σχήμα 1: Συνοπτική παρουσίαση μεθόδων MOR.....	31
--	----



## Κατάλογος Πινάκων

Πίνακας 4.1: Συναρτήσεις για εισαγωγή δεδομένων.....	40
Πίνακας 4.2: Συναρτήσεις για εξαγωγή δεδομένων .....	40
Πίνακας 6.1: Αποτέλεσμα ανάγνωσης αρχείου.....	57
Πίνακας 6.2: MatrixMarket φορμάτ αρχείου.....	58
Πίνακας 7.1: Χαρακτηριστικά του συστήματός μας.....	63
Πίνακας 7.2: Ενδεικτικό Benchmark μικρής διάστασης πίνακα .....	64
Πίνακας 7.3: Χρόνοι εξαγωγής πινάκων και προσομοίωσης.....	73

## Κατάλογος Συντομογραφιών

MOR:	Model Order Reduction
TBR:	Truncated Balanced Realization
MATLAB:	MATrix LABoratory
HSV:	Hankel Singular Values
CAD:	Computer Aided Design
LTI:	Linear Time Invariant
MNA:	Modified Nodal Analysis
NA:	Nodal Analysis
IC:	Integrated Circuit
MIMO:	Multi Input – Multi Output
MM:	MatrixMarket
RAM:	Random Access Memory
KCL:	Kirchoff Current Law
KVL:	Kirchoff Voltage Law
POD:	Proper Orthogonal Decomposition
PR:	Positive Real
VLSI:	Very Large Scale Integration
RHS:	Right Hand Side
CPU:	Central Processing Unit

**Η** χρήση των προσομοιωτών σε ολοκληρωμένα κυκλώματα πολύ μεγάλης κλίμακας, τις περισσότερες φορές οδηγεί στην μοντελοποίηση τους με υπολογιστικά πολύπλοκα μαθηματικά μοντέλα. Αυτό έχει ως απόρροια να δαπανώνται μεγάλα ποσοστά χρόνου, επεξεργαστικής ισχύος και αποθηκευτικού χώρου του μηχανήματος μας μέχρις ότου να λάβουμε την απόκριση συχνότητας του συστήματος.

Η παρούσα διπλωματική εργασία πραγματεύεται τους αλγορίθμους, οι οποίοι στην επιστημονική βιβλιογραφία συναντώνται με το όνομα Model Order Reduction (MOR) και εστιάζει την προσοχή της, υλοποιώντας τον πιο αποδοτικό από αυτούς, τον Truncated Balanced Realization (TBR). Ο στόχος αυτής της κατηγορίας αλγορίθμων είναι να παραγάγουμε ένα νέο μοντέλο συστήματος που περιγράφεται από έναν πίνακα και ένα διάνυσμα κατάστασης, μικρότερης τάξης σε σχέση με τα αρχικά, συνεπώς ευκολότερα διαχειρίσιμα, όπου όμως με το ίδιο διάνυσμα εισόδου, η απόκριση συχνότητας του θα προσεγγίζει όσο το δυνατόν περισσότερο την αντίστοιχη απόκριση του αρχικού μοντέλου. Το νέο μοντέλο συστήματος που θα παραχθεί, θα απαρτίζεται από πίνακες και διανύσματα μικρότερης τάξης και επομένως είναι αντιληπτό ότι θα εξοικονομεί πόρους από το σύστημα μας, τόσο από πλευράς υπολογιστικού κόστους όσο και από απαιτήσεις μνήμης RAM.

Μεταξύ άλλων, έχουμε επιλέξει τον αλγόριθμο TBR προς μελέτη, καθώς βρίσκεται σε πλεονεκτική θέση. Ο συγκεκριμένος αλγόριθμος παρέχει υπολογίσιμα όρια σφάλματος, γεγονός που αποτελεί στοιχείο της απόδοσης και της ορθότητας της λύσης του. Επιπλέον, το νέο, μειωμένης κλίμακας μοντέλο συστήματος που προκύπτει από τον TBR θα είναι εγγυημένα αριθμητικά ευσταθές και θα διατηρεί τις όποιες ιδιότητες (όπως παθητικότητα, συμμετρία) κατείχε το αρχικό.

Ο αλγόριθμος TBR μοντελοποιείται και υλοποιείται στο λογισμικό πακέτο της MATrix LABoratory (MATLAB) χρησιμοποιώντας το ενσωματωμένο εργαλείο «Control System Toolbox». Τα αποτελέσματα που λαμβάνονται από την προσομοίωση του αλγορίθμου παρουσιάζονται για να επιδείξουμε το αλάθητο της μεθόδου, συγκρίνοντας την μικρή απόκλιση της γραφικής παράστασης της συνάρτησης μεταφοράς του νέου, ελαττωμένου σε τάξη συστήματος, με την αντίστοιχη αρχική.

Συμπερασματικά, το καινούριο, μικρότερης κλίμακας δυναμικό σύστημα θα προσεγγίζει την απόκριση συχνότητας του αρχικού, ακατέργαστου συστήματος, φραγμένο πάντα από ένα άνω όριο σφάλματος, αλλά η προσομοίωση θα γίνεται με ευκολότερο και προπαντός γρηγορότερο τρόπο.

## ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Υποβιβασμός τάξης μοντέλου συστήματος; Truncated Balanced Realization; Υπολογίσιμο όριο σφάλματος; Προσομοίωση; Ολοκληρωμένα Κυκλώματα πολύ μεγάλης κλίμακας; Διανύσματα Ελεγκσιμότητας και Παρατηρησιμότητας; Lyapunov Εξισώσεις; Ευστάθεια

## ***ABSTRACT***

**T**he use of simulators in very large scaled integrated circuits in most of the times leads to modelling them with computational complex mathematical models. This has as a result a large waste of time, processing power and storage space until we receive the frequency response of the system.

This thesis deals with the algorithms, which in scientific literature are termed as Model Order Reduction Algorithms and focuses on the implementation of the most effective of them, named Truncated Balanced Realization (TBR). The aim of the algorithms belonging to this category is to produce a new system model that can be described from smaller order array and state vector compared to the initial ones, which as a result will be much more easily handled. The main purpose is with the same input of vector, the response frequency will approach as much as possible the response frequency of the initial system. The new system model produced, will consist of arrays and vectors of lower order and therefore, as it is easily understood, it will save system's resources both in terms of computational cost and from RAM memory requirements.

Among other algorithms, we have chosen to study the TBR algorithm since it is in advantageous position. The specific algorithm provides computable error bounds, which is a significant fact of the efficiency and accuracy of the provided solution. Furthermore, the new, reduced order system model which is resulting from TBR, will be guaranteed numerically stable and will also preserve the same properties (such as passivity, symmetry) that the initial model owned.

The algorithm TBR is modeled and implemented in the software package of MATrix LABoratory (MATLAB) using the embedded tool «Control System Toolbox». The simulation's results are presented in order to show the method's infallibility, comparing the small divergence between the graph of the new transfer function of the reduced order system and the corresponding initial one.

In conclusion, the new, reduced order dynamical system will approach the frequency response of the initial, crude system, always protected from an upper error bound, but the simulation will be implemented in an easier and above all in a much faster way.

## **KEY WORDS**

Model Order Reduction; Truncated Balanced Realization; Computable error bounds; Simulation; Very Large Scale Integrated Circuits(VLSI); Controllability and Observability Grammians; Lyapunov Equations; Stability

*“Ambition is the path to success,  
persistence is the vehicle you arrive in.”*

*-William Eardley IV*

*“The process of scientific discovery is,  
in effect, a continual flight from wonder.”*

*-Albert Einstein*

*“Do your work with your whole heart, and  
you will succeed - there's so little competition.”*

*-Elbert Hubbar*

# ΚΕΦΑΛΑΙΟ 1

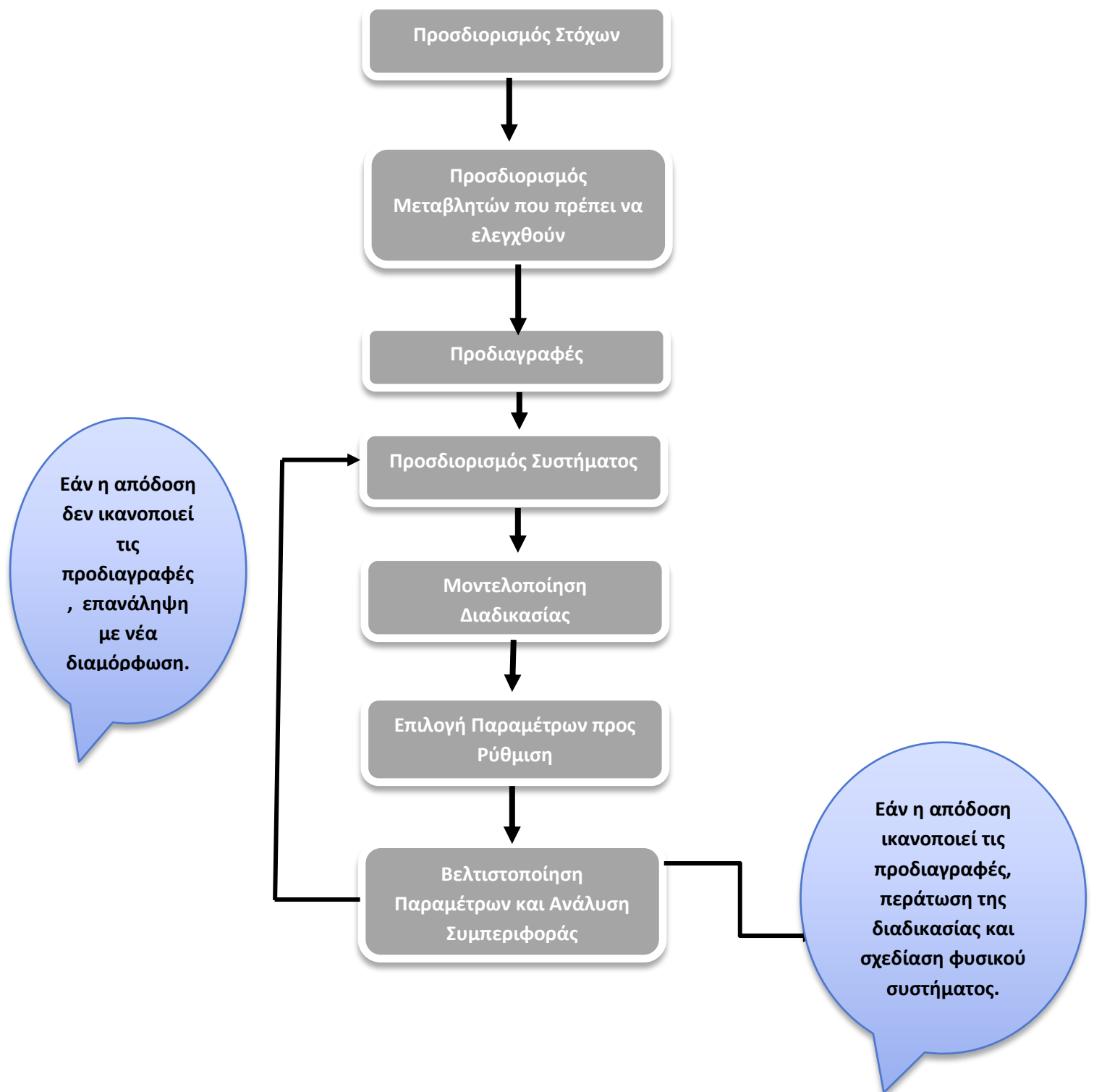
## Εισαγωγή

### 1.1 Προσομοίωση σήμερα

Οι τεχνικές και οι μέθοδοι Προσομοίωσης διαδραματίζουν έναν ουσιαστικό ρόλο στην επιστήμη, πλέον, αφού χρησιμοποιούνται για να προσεγγίσουν την συμπεριφορά συστημάτων που συναντώνται σε μια ευρεία γκάμα διαφορετικών επιστημονικών πεδίων, όπως είναι η Βιοϊατρική Μηχανική, η Αεροναυτική, τα Συστήματα Αυτομάτου Ελέγχου πτήσεων καθώς και τα Συστήματα Παραγωγής και Διανομής Ηλεκτρικής Ενέργειας.

Πιο αναλυτικά, με την βοήθεια των σύγχρονων εργαλείων προσομοίωσης μπορούμε να κατασκευάσουμε μια εικονική μορφή του συστήματος που θα θέλαμε να παραγάγουμε στην πραγματικότητα και στην συνέχεια να το ελέγξουμε και να διαπιστώσουμε κατά πόσο η συμπεριφορά του ανταποκρίνεται στις αρχικές προδιαγραφές του προβλήματός μας (Εικόνα 1). Ο λόγος για τον οποίο συχνά η σύγχρονη βιομηχανία πρώτα προσομοιώνει τα συστήματα της προτού τα παράγει είναι ότι στην περίπτωση που δεν ανταποκρίνονται στις αρχικές προδιαγραφές που θέτει το πρόβλημα προς επίλυση, δεν θα έχει επενδύσει άσκοπα χρόνο, χρήματα και υλικά για την ρεαλιστική ανάπτυξη του συγκεκριμένου συστήματος. Αρκετές φορές, μάλιστα, ο πειραματισμός με πραγματικά συστήματα και περιβάλλοντα είναι ιδιαίτερα επικίνδυνος (π.χ. το σύστημα ενός πυρηνικού αντιδραστήρα). Αντίθετα, μέσω της προσομοίωσης, θα αναγνωρίσουμε τα ισχυρά και τα αδύναμα σημεία του εικονικού μοντέλου, θα αναδιατυπώσουμε τη σχεδιαστική πρόταση, θα το δοκιμάσουμε ξανά και θα προχωρήσουμε εν τέλει στην παραγωγή ενός επιτυχούς φυσικού μοντέλου. Υπάρχουν, ωστόσο διαδικασίες προσομοίωσης συστημάτων που μπορεί να διαρκέσουν ακόμη και ολόκληρους μήνες. Εάν όμως αναλογιστούμε την ζημία που θα είχε η συγκεκριμένη επιχείρηση, όταν μετά την ρεαλιστική παραγωγή του προϊόντος της, διαπίστωνε την αναξιοπιστία του, θα συμπεράνουμε την χρηστικότητα των προσομοιώσεων σήμερα.

Τις τελευταίες δεκαετίες, φαίνεται πως οι εφαρμογές προσομοίωσης έχουν κερδίσει έδαφος στον τομέα της επιστήμης των υπολογιστών και κυρίως εκείνες που στρέφονται στα ολοκληρωμένα κυκλώματα μεγάλης κλίμακας υπολογιστών (VLSI, Very Large Scale Integration).



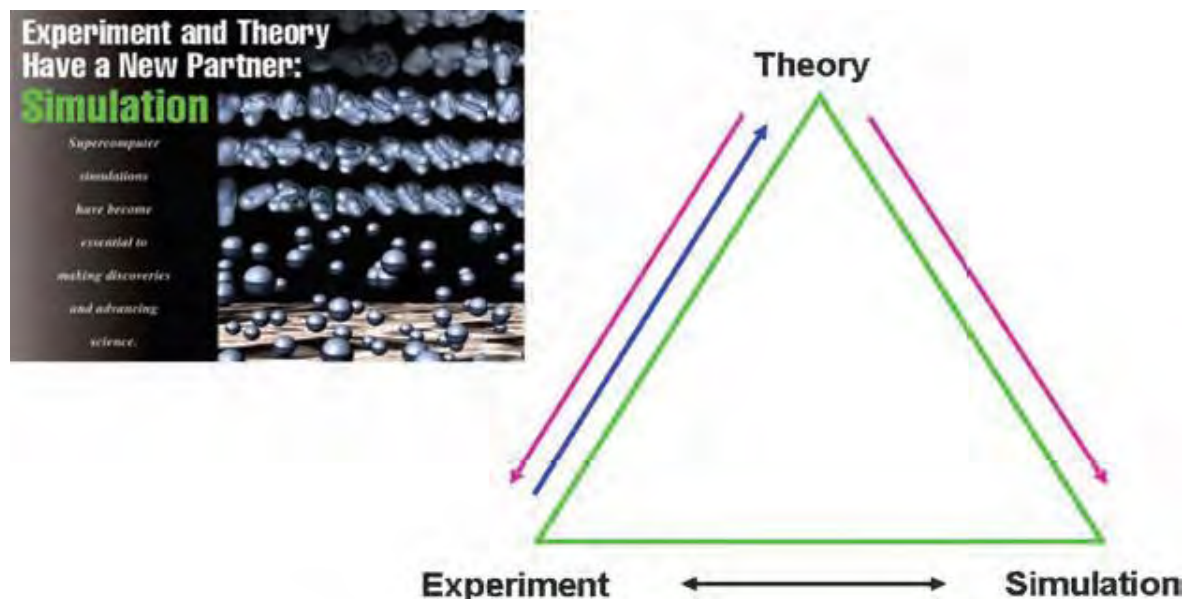
Εικόνα 1: Διαδικασία σχεδίασης ενός φυσικού συστήματος

Στην συνέχεια, παρουσιάζουμε δύο βαρυσήμαντους ορισμούς για το σύνολο αυτής της διπλωματικής εργασίας, της *μοντελοποίησης* και της *προσομοίωσης*.

1. **Προσομοίωση:** Η μίμηση μιας πραγματικής ή πιθανής κατάστασης ή ενός γεγονότος, με σκοπό την εύρεση μιας παρελθοντικής αιτίας ή την πρόβλεψη μιας μελλοντικής συμπεριφοράς. Η προσομοίωση μπορεί να εκτελεστεί μέσω:
  - της επίλυσης μιας σειράς εξισώσεων, δηλαδή μέσω ενός μαθηματικού μοντέλου
  - της κατασκευής ενός φυσικού μοντέλου σε κλίμακα
  - σταδιακών δοκιμών
  - γραφικών υπολογιστικών μοντέλων όπως είναι τα διαγράμματα ροής.
2. **Μοντελοποίηση:** Η γραφική, μαθηματική (συμβολική), φυσική αναπαράσταση ή η απλοποιημένη εκδοχή του σχεδίου ενός φαινομένου, μιας δομής, ενός συστήματος ή μιας πτυχής του πραγματικού κόσμου. Οι στόχοι ενός μοντέλου περιλαμβάνουν:
  - την κατανόηση του εν λόγω συστήματος, εξαλείφοντας τα περιττά συστατικά
  - τη διευκόλυνση στη λήψη αποφάσεων μέσα από την προσομοίωση διαφορετικών σεναρίων
  - την πρόβλεψη και τον έλεγχο μελλοντικών γεγονότων με βάση παρελθοντικές παρατηρήσεις

Η διακύμανση των ειδών των μοντέλων είναι μεγάλη, εφόσον μοντέλο μπορεί να θεωρηθεί ένα απλό σκίτσο ή ένα υπολογιστικό πρόγραμμα χιλιάδων γραμμών κώδικα.

Κατά συνέπεια, παρατηρούμε ότι οι επιτακτικές ανάγκες της βιομηχανίας έχουν οδηγήσει στην εδραίωση μιας τρίτης μετρικής, πέρα από τις κλασσικές, την θεωρητική και την πειραματική, την προσομοίωση (Εικόνα 2). Η προσομοίωση σήμερα αποτελεί τον συνδετικό κρίκο και το ενδιάμεσο στάδιο μεταξύ θεωρίας και πειράματος [8].



Εικόνα 2: Προσομοίωση - συνδετικός κρίκος θεωρίας και πειράματος



## 1.2 Περιγραφή του Προβλήματος και Συμβολή της Διπλωματικής Εργασίας

Τα τελευταία χρόνια με δεδομένη την ραγδαία εξέλιξη της τεχνολογίας, της τεχνογνωσίας και του εξειδικευμένου ανθρώπινου δυναμικού στο ευρύτερο πεδίο της επιστήμης των υπολογιστών, τα ολοκληρωμένα κυκλώματα των υπολογιστικών μονάδων που θα πρέπει να προσομοιωθούν προτού κατασκευαστούν γίνονται ολοένα και πιο πολύπλοκα και κατ' επέκταση δύσκολα στον χειρισμό τους.

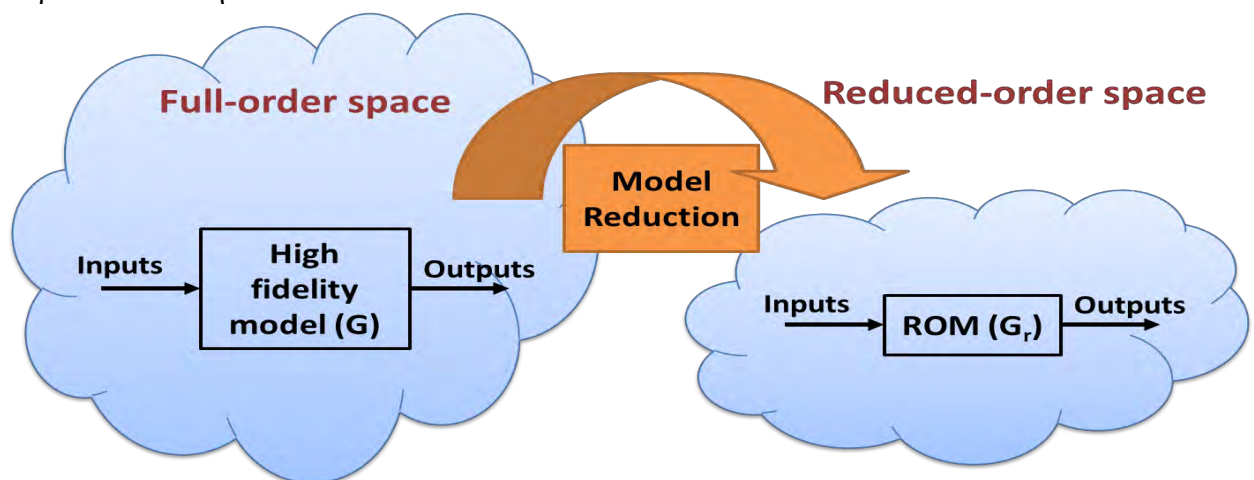
Παρατηρούμε ότι είναι πολύ πιθανό να έχουμε να προσομοιώσουμε ένα κύκλωμα με πολύ μεγάλο αριθμό καταστάσεων ή κόμβων, έστω  $n$ , και μικρό αριθμό εισόδων ή εξόδων, έστω  $m$ , συγκριτικά με τους κόμβους του συστήματος, δηλαδή να ισχύει  $m \ll n$ . Έτσι, για την εγγυημένη ακρίβεια και την διευκόλυνση της μοντελοποίησης των υπολογισμών έχουν αναπτυχθεί στην επιστημονική βιβλιογραφία τα εργαλεία Computer Aided Design (CAD) για το ευρύτερο σύνολο των μηχανικών. Η μοντελοποίηση και προσομοίωση των κυκλωμάτων που πραγματοποιείται με την βοήθεια αυτών των υπολογιστικών εργαλείων όμως οδηγεί, αφενός, σε ακριβή, αφετέρου δε, σε υπολογιστικά πολύπλοκα και χρονοβόρα μοντέλα, τα οποία τις περισσότερες φορές περιγράφονται από υψηλής τάξης διαφορικές εξισώσεις. Γίνεται αντιληπτό, λοιπόν, ότι η προσομοίωση και η ανάλυση τόσο μεγάλης κλίμακας κυκλωμάτων είναι ιδιαίτερα δαπανηρή από πλευράς υπολογιστικού κόστους. Οι διάφορες πράξεις που θα επενεργήσουν πάνω στον μεγάλης διάστασης πίνακα καταστάσεων, θα απαιτήσουν πολλούς πόρους από το υπολογιστικό μας σύστημα, με έμφαση στην μνήμη RAM και στην χρήση του επεξεργαστή - CPU [7]. Σαν μηχανικοί λοιπόν καλούμαστε να βρούμε τρόπο και λύση ώστε η προσομοίωση κυκλωμάτων να γίνεται οικονομικότερα, γρηγορότερα και μέσα σε πραγματικούς χρόνους.

Ένα φλέγον ζήτημα που απασχολεί τις τελευταίες δεκαετίες έντονα την ερευνητική κοινότητα τόσο των πανεπιστημιακών ιδρυμάτων όσο και της βιομηχανίας (Research & Development Companies) είναι η ανακάλυψη καινοτόμων μεθόδων και τεχνικών μείωσης της τάξης μεγέθους των πολύ μεγάλης διάστασης πινάκων που περιγράφουν ολοκληρωμένα κυκλώματα. Είναι πολύ χρήσιμο να μελετάμε σύνθετα συστήματα υψηλής τάξης, χρησιμοποιώντας προσεγγιστικά μοντέλα χαμηλότερης τάξης. Προκειμένου, λοιπόν, να ανταποκριθούμε σε μια τέτοια πρόκληση και να παράξουμε πίνακες μικρότερης διάστασης και άρα ευκολότερα διαχειρίσιμους, έχει αναπτυχθεί μια ολόκληρη γκάμα αλγορίθμων οι οποίοι συναντώνται στην επιστημονική βιβλιογραφία των Συστημάτων Αυτομάτου Ελέγχου με το όνομα Model Order Reduction Algorithms (MOR). Ο στόχος είναι να παράξουμε ένα νέο μοντέλο κυκλώματος που περιγράφεται από έναν πίνακα κατάστασης μικρότερης τάξης σε σχέση με τον αρχικό, όπου όμως με το ίδιο διάνυσμα εισόδου, η απόκριση συχνότητας του να προσεγγίζει όσο το δυνατόν περισσότερο την αντίστοιχη απόκριση του αρχικού μοντέλου, έχοντας ταυτοχρόνως μικρότερες απαιτήσεις αποθηκευτικού χώρου και υπολογιστικού κόστους (Εικόνα 3).

Η παρούσα διπλωματική εργασία πραγματεύεται τους αλγορίθμους υποβιβασμού τάξης μοντέλου (Model Order Reduction, MOR) για ολοκληρωμένα

κυκλώματα και συστήματα υπολογιστών με πολύ μεγάλο αριθμό καταστάσεων και υλοποιεί έναν εξ' αυτών, τον Truncated Balanced Realization (TBR). Στα πλαίσια της μελέτης που διεξάγουμε επιλέχθηκε για περαιτέρω ανάλυση και προσομοίωση ο αλγόριθμος TBR, ο οποίος είναι κατάλληλος για τον υπολογισμό μειωμένης τάξης μοντέλων γραμμικά ανεξάρτητων συστημάτων, αφού διακρίνεται για την ιδιαίτερη συμπεριφορά του μεταξύ των υπολοίπων. Καταρχήν, η περάτωση του αλγορίθμου εγγυάται ότι το νέο μοντέλο μειωμένης τάξης θα είναι με βεβαιότητα ευσταθές, δηλαδή, δεν θα παρουσιάζει απρόβλεπτη συμπεριφορά στις υψηλές συχνότητες και η νέα συνάρτηση μεταφοράς σε σχέση με την συνάρτηση μεταφοράς του αρχικού συστήματος θα είναι άνω φραγμένη από μια αριθμητικά υπολογίσιμη ποσότητα. Με άλλα λόγια, ο αλγόριθμος προσφέρει υπολογίσιμα όρια λάθους ανάμεσα στο αρχικό και στο τελικό, μικρότερης διάστασης σύστημα, γεγονός που μας δηλώνει κατά πόσο προσεγγίζει η απόκριση του νέου συστήματος την αντίστοιχη του αρχικού. Αποτελεί, δηλαδή ένα στοιχείο της απόδοσης και αποτελεσματικότητας της ακρίβειας του.

Η βασική ιδέα του αλγορίθμου έγκειται στον υπολογισμό ενός μετασχηματισμού ομοιότητας, ο οποίος επιδρά και μετασχηματίζει τους πίνακες του αρχικού συστήματος, απαιτώντας τα Grammians διανύσματα ελεγκσιμότητας και παρατηρησιμότητας να είναι ίσα και διαγώνια ταυτοχρόνως. Ακολουθώντας στο κείμενο πρόκειται να παρουσιάσουμε την φυσική ερμηνεία και την διαίσθηση που κρύβεται πίσω από αυτήν την απαίτηση. Ενδεικτικά, αναφέρουμε εδώ ότι επιθυμούμε να εξαλείψουμε τις καταστάσεις, οι οποίες είναι είτε μη ελέγξιμες (non - Controllable), είτε μη παρατηρήσιμες (non - Observable), είτε και τα δύο. Θεωρούμε, ότι το σύστημα κάτω από εξαιρετικά δύσκολες συνθήκες θα οδηγηθεί σε αυτές τις καταστάσεις (ίσως και ποτέ) και άρα δεν ωφελεί σε τίποτα στο να τις συμπεριλαμβάνουμε στο σύστημα προς ανάλυση και να το επιβαρύνουμε υπολογιστικά. Ο αριθμητικός υπολογισμός των Controllability και Observability Grammians βασίζεται στην επίλυση των δυικών Lyapunov εξισώσεων. Το νέο, μειωμένης διάστασης μοντέλο συστήματος λαμβάνεται, αφού πρώτα διαμερίσουμε το αρχικό σε ισχυρά και αδύναμα υποσυστήματα και στην συνέχεια περικόσουμε κατάλληλα τα αδύναμα υποσυστήματα που διαθέτουν την παραπάνω ιδιότητα.

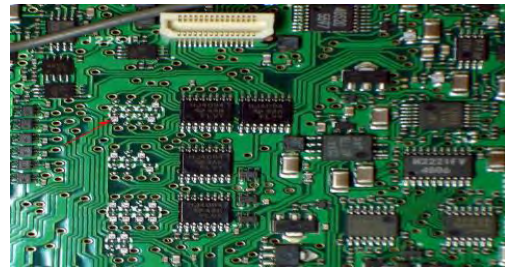


Εικόνα 3: Ενδεικτική Μορφή Model Order Reduction

## 1.3 Ιστορική Αναδρομή

Ήδη από τα πολύ παλιά χρόνια, όταν ακόμη η επιστήμη των υπολογιστών δεν είχε αναπτυχθεί, υπήρξε η επιθυμία στον κλάδο των Μαθηματικών να μειώσουμε τα μεγέθη των μοντέλων προς επίλυση, επιτυγχάνοντας κατ' αυτόν τον τρόπο ταχύτερους υπολογισμούς λύσεων, χωρίς όμως τα νέα να υστερούν σε ακρίβεια. Για παράδειγμα, η πρώτη απόπειρα για μεθόδους MOR έγινε το 1807 από τον Fourier, ο οποίος δημοσίευσε το άρθρο του για την προσέγγιση συναρτήσεων με λίγους τριγωνομετρικούς όρους. Λίγα χρόνια αργότερα, περί το 1893-1974, ο Lanczos προσπάθησε να μειώσει την διάσταση του προβλήματος που είχε να επιλύσει μετατρέποντας τους πίνακες σε άνω τριγωνικούς και εξαλείφοντας τα άχρηστα μέρη τους [8].

Τα ολοκληρωμένα κυκλώματα (Integrated Circuit - IC) υπολογιστών εμφανίστηκαν περίπου το 1960 και μέρα με την μέρα η πολυπλοκότητα τους αυξανόταν με εκθετικό ρυθμό (Εικόνα 4).



Εικόνα 4: Σύγχρονη μορφή ολοκληρωμένων κυκλωμάτων

Από την άλλη, οι πρώτες αξιόλογες προσπάθειες για ανάπτυξη αλγορίθμων MOR στο πεδίο της Θεωρίας Συστημάτων Ελέγχου ξεκίνησαν περίπου στο 1980. Λίγο νωρίτερα, γύρω στο 1976 οι Mullis<sup>1</sup> και Robert εισήγαγαν για πρώτη φορά στην ιστορία τον αλγόριθμο Truncated Balanced Realization. Το 1981 ο Moore δημοσίευσε επίσημα τον αλγόριθμο, ενώ το 1984 ο Glover δημοσίευσε την ερευνητική δουλειά του για την εύρεση των Hankel Norms. Το 1990 δημοσιεύτηκαν τα ευρήματα για την διαδικασία υπολογισμού των υποχωρών Krylov. Συνεπώς, όλες οι προαναφερθέντες μέθοδοι, βρίσκουν ευρεία απήχηση και εφαρμογή από τότε μέχρι και σήμερα, στην μείωση μεγέθους συστημάτων των ολοκληρωμένων κυκλωμάτων πολύ μεγάλης κλίμακας.

## 1.4 Αποτελέσματα και Σκοπός της Διπλωματικής Εργασίας

Η σκοπιμότητα της έρευνας που διεξάγεται μέσω αυτής της διπλωματικής εργασίας είναι μέσω των αλγορίθμων που στοχεύουν στην εύρεση μοντέλων συστημάτων μικρότερης τάξης, να παραγάγουμε ισοδύναμα δυναμικά συστήματα με τα αρχικά, τα οποία δεν θα αποκλίνουν σημαντικά από την απόκριση συχνότητας που δίδει το αρχικό

---

<sup>1</sup> **Kary Banks Mullis** γεννήθηκε το 1944 και υπήρξε βραβευμένος με βραβείο Νόμπελ. Αμερικανός βιοχημικός, συγγραφέας και λέκτορας.

μοντέλο, θα είναι αριθμητικά ευσταθή και θα διατηρούν τις ιδιότητες του αρχικού συστήματος.

Επιπλέον, σκοπός της μελέτης είναι να οριστεί ένα άνω όριο φράγματος λάθους της συνάρτησης μεταφοράς μεταξύ του καινούριου και του αρχικού συστήματος. Παράλληλα, ο αλγόριθμος TBR που θα υλοποιήσουμε θα πρέπει να ολοκληρώνεται, εξοικονομώντας σημαντικά ποσοστά μνήμης και χρόνου επεξεργασίας εν συγκρίσει με το πρωταρχικό μοντέλο συστήματος. Η εξοικονόμηση επεξεργαστικής ισχύος είναι ως ένα βαθμό αναμενόμενη, εφόσον πλέον έχουμε να επιλύσουμε το σύστημα δυναμικών καταστάσεων για μικρότερης διάστασης πίνακες, άρα συντομότερα και ταχύτερα. Επίσης, έχουμε να χειριστούμε μικρότερης διάστασης σύστημα και επομένως θα καταλαμβάνει λιγότερο αποθηκευτικό χώρο στην μνήμη.

Τελευταίος στόχος, όχι όμως επουσιώδους σημασίας που τίθεται, είναι ο αλγόριθμος TBR, να παράγει νέα, μειωμένα μοντέλα δυναμικών συστημάτων τα οποία θα συμπεριλαμβάνουν μόνο τις καταστάσεις που είναι ταυτόχρονα ελέγξιμες και παρατηρήσιμες και να περικόψει όσες δεν ανήκουν είτε στην μία είτε στην άλλη κατηγορία.

## 1.5 Διάρθρωση της Διπλωματικής Εργασίας

Η δομή της παρούσης μελέτης κινείται σε 3 βασικούς άξονες. Στο 1ο μέρος, όπου συμπεριλαμβάνονται τα κεφάλαια 2 και 3 παρουσιάζεται η μελέτη των θεωρητικών μαθηματικών και αλγόριθμοι παρόμοιας σκοπιμότητας με του αλγορίθμου TBR. Στο 2ο μέρος, όπου εμπεριέχονται τα κεφάλαια 4, 5 και 6 γίνεται η ανάλυση των βημάτων του αλγορίθμου TBR και της προσομοίωσης του στο περιβάλλον ανάπτυξης MATLAB. Στην 3η και τελευταία γραμμή οργάνωσης της εργασίας μας, δηλαδή στα κεφάλαια 7 και 8 παρουσιάζονται και αναλύονται τα προκύπτοντα αριθμητικά αποτελέσματα της παρούσης διατριβής και γίνονται σκέψεις για μελλοντικές επεκτάσεις. Στην συνέχεια, παρουσιάζουμε περιγραφικά τα βασικά στοιχεία κάθε κεφαλαίου.

- **Στο κεφάλαιο 2**, παρουσιάζεται το μαθηματικό υπόβαθρο, δηλαδή οι έννοιες της γραμμικής άλγεβρας και των συστημάτων αυτομάτου ελέγχου, που θα πρέπει να φρεσκάρει στην μνήμη του ο αναγνώστης, ώστε να κατανοήσει πλήρως τον κεντρικό άξονα της μεθόδου αλλά και όλες τις αλγοριθμικές λεπτομέρειες της υλοποίησης.
- **Στο κεφάλαιο 3**, γίνεται μια σύντομη ανασκόπηση στην επιστημονική βιβλιογραφία των πιο διαδεδομένων μεθόδων εύρεσης μειωμένης τάξης μοντέλου, τα προτερήματα και τις αδυναμίες τους, καθώς και μια εκτενής ανάλυση των ιδιοτήτων και των γνωρισμάτων της επικείμενης μεθόδου μας. Παράλληλα, δίδεται έμφαση στην φυσική και διαισθητική ερμηνεία της μεθόδου.

- **Στο κεφάλαιο 4**, πραγματοποιείται μία περιγραφή του περιβάλλοντος ανάπτυξης που χρησιμοποιήθηκε για την μοντελοποίηση και την προσομοίωση του κυκλώματος. Συγχρόνως, περιγράφουμε εν συντομία το γραφικό περιβάλλον του λογισμικού της MATLAB και του ενσωματωμένου εργαλείου “Control System Toolbox”, τις δυνατότητες που προσφέρουν στο ευρύτερο πεδίο του μηχανικού, αλλά και τα είδη αρχείων που υποστηρίζουν, όπως τα m-Files (Scripts - Functions).
- **Στο κεφάλαιο 5**, γίνεται εκτενής αναφορά στα βήματα του αλγορίθμου TBR. Επίσης, περιγράφεται μια παραλλαγή της βασικής εκδοχής του αλγορίθμου για επίτευξη καλών προσεγγίσεων στις χαμηλές συχνότητες (dc frequencies).
- **Στο κεφάλαιο 6**, παραθέτουμε το τρόπο και τα βήματα με τον οποίο γίνεται μια τυπική προσομοίωση κυκλώματος, καθώς και την αναλυτική μεθοδολογία που ακολουθήθηκε για την προσομοίωση του αλγορίθμου εύρεσης μοντέλου μειωμένης τάξης, αλλά με απόκριση που προσεγγίζει όσο το δυνατόν περισσότερο την απόκριση του αρχικού συστήματος, TBR. Παράλληλα, αναφερόμαστε στην απαιτούμενη μορφή που θα πρέπει να έχει το αρχείο εισόδου που εμπεριέχει την περιγραφή του κυκλώματος και τον τρόπο με τον οποίο θα μετατραπεί σε μορφή πίνακα, ώστε να είμαστε σε θέση να επεμβούμε στο κύκλωμα μέσω του πίνακα κατάστασης και να μειώσουμε την διάσταση του με την βοήθεια μετασχηματισμών ομοιότητας.
- **Στο κεφάλαιο 7**, παρουσιάζουμε τα αποτελέσματα της εκτέλεσης του αλγορίθμου Truncated Balanced Realization, την συγκριτική μελέτη της διάστασης του νέου, μειωμένου μοντέλου συστήματος με σημείο αναφοράς το αρχικό μοντέλο, καθώς και κατά πόσο ανταποκρίνεται η πρακτική υλοποίηση του αλγορίθμου στην θεωρητική προσέγγιση. Επιπλέον, δίνουμε κάποια ενδεικτικά screenshots της προσομοίωσης που υποδεικνύουν την εγκυρότητα του Model Order Reduction Αλγορίθμου μας ως προς την ακρίβεια της λύσης, μιας και η γραφική παράσταση της καινούριας συνάρτησης μεταφοράς τείνει να ταυτιστεί με την αρχική συνάρτηση μεταφοράς. Το τελευταίο υποδεικνύει ότι παρά το μικρότερης διάστασης σύστημα, το οποίο χρησιμοποιήσαμε για τον ταχύτερο υπολογισμό της λύσης, η απόκριση του νέου συστήματος, με δεδομένο διάλυσμα εισόδου, φράσσεται από ένα ανεκτό και επιτρεπτό όριο σφάλματος.
- **Στο κεφάλαιο 8**, γίνεται μια σύνοψη της έρευνας που διεξήγαμε, διατυπώνονται συμπεράσματα αναφορικά με τα αποτελέσματα που ελήφθησαν από την υλοποίηση και προτείνονται ενδεχόμενες μελλοντικές προεκτάσεις ή παραλλαγές της βασικής εκδοχής του αλγορίθμου TBR.

Η σελίδα αυτή είναι σκόπιμα λευκή.

# ΚΕΦΑΛΑΙΟ 2

## Μαθηματικό Υπόβαθρο

Η ανάπτυξη ενός ορθού μαθηματικού μοντέλου για την περιγραφή ενός φυσικού συστήματος διαδραματίζει καταλυτικό ρόλο στην θεωρία Συστημάτων Αυτομάτου Ελέγχου [3]. Στην ενότητα αυτή, θα περιγράψουμε αναλυτικά όλους τους ορισμούς από το πεδίο των Μαθηματικών και πιο ειδικά της Γραμμικής Άλγεβρας, ώστε να υπάρχουν όλα τα απαραίτητα εφόδια για να γίνει κατανοητή η μοντελοποίηση του αλγορίθμου TBR.

### 2.1 Κατηγορίες Συστημάτων

Η δημιουργία ενός μοντέλου για την περιγραφή ενός φυσικού συστήματος είναι διπλής σημασίας. Πρώτον, μας βοηθά στην βαθύτερη κατανόηση και αφομοίωση των αρχών που διέπουν την συμπεριφορά του συστήματος και δεύτερον μας δίνει την ευκαιρία να μελετήσουμε την συμπεριφορά του σε διάφορες συνθήκες μέσω προσομοιώσεων.

Τα συστήματα που προκύπτουν από μια τέτοια μοντελοποίηση διακρίνονται σε 2 μεγάλες κατηγορίες, τα στατικά και τα δυναμικά. Εμείς στην παρούσα μελέτη θα χρησιμοποιήσουμε τα δυναμικά συστήματα, ωστόσο όμως σε αυτό το κεφάλαιο και για λόγους πληρότητας της παρουσίασης θα ορίσουμε και τα δύο είδη συστημάτων. Ένα σύστημα λέγεται **στατικό** ή σύστημα δίχως μνήμη [3], όταν για κάθε χρονική στιγμή  $t$ , το σήμα εξόδου  $y(t)$ , εξαρτάται μόνο από την τιμή του σήματος εισόδου  $x(t)$ , την ίδια χρονική στιγμή  $t$ . Χαρακτηριστικό παράδειγμα αποτελεί η ωμική αντίσταση  $R$ . Απεναντίας, ένα σύστημα καλείται **δυναμικό** [3], όταν η έξοδος του  $y(t)$ , την χρονική στιγμή  $t$ , εξαρτάται από τις τιμές που παίρνει η είσοδος και σε χρονικές στιγμές διάφορες του  $t$ , όπως σε ένα διάστημα  $[t_0, t]$ . Τα δυναμικά συστήματα έχουν μνήμη και σε αυτά συγκαταλέγεται, λόγου χάριν, η χωρητική αντίσταση,  $C$ .

Τα δυναμικά συστήματα με την σειρά τους διαχωρίζονται σε γραμμικά και σε μη-γραμμικά. Εισαγωγικά, επισημαίνουμε ότι ένα σύστημα θα βρίσκεται σε ηρεμία, εάν αυτό δεν έχει υποστεί καμιά διέγερση από οποιοδήποτε σήμα εκτός του διαστήματος  $[t_0, t_1]$ , κατά την διάρκεια του οποίου το σύστημα διεγείρεται από το  $x(t)$ . Έτσι, ένα σύστημα θα ονομάζεται **γραμμικό** [3] όταν με δεδομένα 2 σήματα  $x_1(t)$ ,  $x_2(t)$  ισχύει:

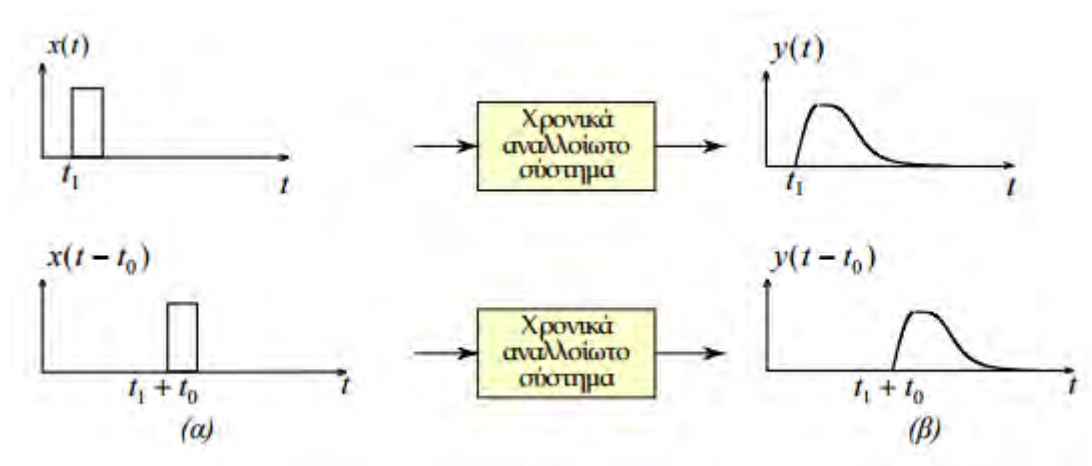
$$F[a_1*x_1(t) + a_2*x_2(t)] = a_1*F[x_1(t)] + a_2*F[x_2(t)] \quad (2.1)$$

για οποιεσδήποτε σταθερές  $a_1, a_2$ . Με άλλα λόγια, η απόκριση του γραμμικού συνδυασμού 2 εισόδων ισοδυναμεί με τον γραμμικό συνδυασμό των επιμέρους αποκρίσεων. Προφανώς, η σχέση (2.1) γενικεύεται για οποιονδήποτε γραμμικό συνδυασμό πεπερασμένου αριθμού σημάτων εισόδων. **Μη - γραμμικό** [3] σύστημα ονομάζεται το σύστημα το οποίο είναι σε ηρεμία, αλλά δεν υπακούει στην σχέση υπέρθεσης (2.1).

Μια άλλη διάκριση των συστημάτων είναι σε χρονικά αμετάβλητα και σε χρονικά μεταβαλλόμενα. Ένα σύστημα θα καλείται **χρονικά αμετάβλητο** [3] όταν και μόνο όταν, οι χρονικές ολισθήσεις του σήματος εισόδου αντιστοιχούν και σε χρονικές ολισθήσεις του σήματος εξόδου (Εικόνα 5). Η πρόταση αυτή μεταφράζεται με μαθηματικό τρόπο ως εξής:

$$y(t) = F[x(t)] \Leftrightarrow y(t-t_0) = F[x(t-t_0)], \quad \text{για κάθε } t_0 \in \mathbb{R} \quad (2.2)$$

Με άλλα λόγια, το παραπάνω σημαίνει ότι το σήμα εξόδου δεν αλλάζει μορφή, ανεξάρτητα από το ποιά χρονική στιγμή θα διεγείρουμε την είσοδο με το σήμα  $x(t)$ . Ένα σύστημα θα ονομάζεται **χρονικά μεταβαλλόμενο** [3], όταν μια ενδεχόμενη χρονική ολίσθηση της εισόδου δεν οδηγήσει σε αντίστοιχη χρονική ολίσθηση της εξόδου, αλλά αντιθέτως προκύψει ένα εντελώς διαφορετικό σήμα εξόδου.



Εικόνα 5: Παράδειγμα εισόδου και εξόδου ΓΧΑ συστήματος:

Το ενδιαφέρον μας στην παρούσα διπλωματική εργασία τόσο από θεωρητικής όσο και από πρακτικής σκοπιάς θα επικεντρωθεί στα δυναμικά Γραμμικά Χρονικά Αμετάβλητα (Linear Time Invariant - LTI) Συστήματα. Αυτά, μετασχηματίζονται από φυσικά σε μαθηματικά μοντέλα με την βοήθεια διαφορικών εξισώσεων.



## 2.2 Χώρος Κατάστασης

### 2.2.1 Ορισμοί Δυναμικών Συστημάτων

Στην συνέχεια θα παρουσιάσουμε μερικούς από τους βαρυσήμαντους ορισμούς των δυναμικών συστημάτων, όπως είναι η έννοια κατάστασης, το διάνυσμα κατάστασης και οι δυναμικές εξισώσεις, καθώς είναι τα θεμέλια πάνω στα οποία θα στηριχτούμε σε επόμενο κεφάλαιο για να αναπτύξουμε τον αλγόριθμό μας [3]. Όλες αυτές οι έννοιες εντάσσονται στην ευρύτερη κατηγορία του χώρου κατάστασης, ο οποίος μας δίνει επιπλέον πληροφορίες για την εσωτερική δομή του υπό εξέταση συστήματος.

➤ Έννοια της κατάστασης

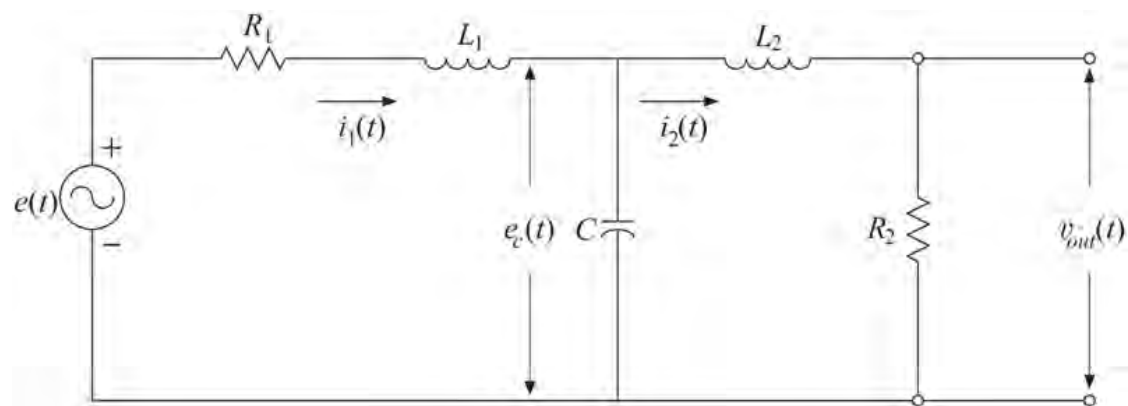
Ως κατάσταση (state) του συστήματος την χρονική στιγμή  $t_0$ , ορίζουμε το σύνολο της ελάχιστης πληροφορίας που θα πρέπει να έχουμε την χρονική στιγμή  $t_0$ , η οποία μαζί με την γνώση της εισόδου  $x(t)$ ,  $t \in [t_0, \infty]$ , καθορίζει πλήρως την συμπεριφορά του συστήματος για  $t \geq t_0$ . Ως συμπεριφορά του συστήματος εννοούμε τις μεταβολές των σημάτων εξόδου και κατάστασης σαν συνάρτηση του χρόνου.

➤ Μεταβλητές / Διάνυσμα Κατάστασης

Ως μεταβλητές κατάστασης ορίζουμε το σύνολο των μεταβλητών που καθορίζουν την κατάσταση ενός συστήματος. Θεωρούμε ότι αποτελούν τις συνιστώσες ενός διανύσματος που ορίζεται ως διάνυσμα κατάστασης.

➤ Δυναμικές Εξισώσεις

Θα προσπαθήσουμε να ορίσουμε τις δυναμικές εξισώσεις με την βοήθεια ενός χαρακτηριστικού παραδείγματος. Έστω ότι διαθέτουμε ένα κύκλωμα της μορφής που φαίνεται στην Εικόνα 6:



Εικόνα 6: Παράδειγμα ενδεικτικού κυκλώματος RLC

- ✓ Στο παραπάνω κύκλωμα επιζητούμε την τάση εξόδου  $V_{out}$  στα άκρα της αντίστασης  $R_2$ .
- ✓ Το διάνυσμα κατάστασης  $\underline{x}(t)$  θα έχει ως μεταβλητές κατάστασης τα ρεύματα  $i_1(t)$ ,  $i_2(t)$  και την τάση  $e_c(t)$  στα άκρα του πυκνωτή, δηλαδή θα είναι:

$$\underline{x}(t) = \begin{bmatrix} i_1(t) \\ i_2(t) \\ e_c(t) \end{bmatrix}$$

Από την εφαρμογή των νόμων του Kirchhoff στο παραπάνω κύκλωμα θα έχουμε:

$$L_1 \frac{di_1(t)}{dt} = -R_1 i_1(t) - e_c(t) + e(t) \quad (2.3)$$

$$L_2 \frac{di_2(t)}{dt} = -R_2 i_2(t) + e_c(t) \quad (2.4)$$

$$C \frac{de_c(t)}{dt} = i_1(t) - i_2(t) \quad (2.5)$$

$$V_{out}(t) = i_2(t) R_2 \quad (2.6)$$

Οι 3 πρώτες από τις παραπάνω εξισώσεις μας περιγράφουν την εξέλιξη των μεταβλητών κατάστασης στον χρόνο, ενώ η τέταρτη την έξοδο του συστήματος συναρτήσει μιας εκ των μεταβλητών κατάστασης. Μετά από μια μικρή αναδιάθρωση των εξισώσεων (2.3), (2.4), (2.5), (2.6) προκύπτει η παρακάτω μορφή εκφρασμένη με την βοήθεια πινάκων και διανυσμάτων:

$$\begin{bmatrix} \frac{di_1(t)}{dt} \\ \frac{di_2(t)}{dt} \\ \frac{de_c(t)}{dt} \end{bmatrix} = \begin{bmatrix} \frac{-R_1}{L_1} & 0 & \frac{-1}{L_1} \\ 0 & \frac{-R_2}{L_2} & \frac{1}{L_2} \\ \frac{1}{C} & \frac{-1}{C} & 0 \end{bmatrix} \begin{bmatrix} i_1(t) \\ i_2(t) \\ e_c(t) \end{bmatrix} + \frac{1}{L_1} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} e(t) \quad (2.7)$$

$$V_{out}(t) = [0 \quad R_2 \quad 0] \begin{bmatrix} i_1(t) \\ i_2(t) \\ e_c(t) \end{bmatrix} \quad (2.8)$$

Έτσι, σε μια πιο γενική μορφή οι σχέσεις (2.7), (2.8) γράφονται ως ακολούθως:

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{b} e(t) \quad (2.9)$$

$$V_{out}(t) = \underline{c}^T \underline{x}(t) \quad (2.10)$$

Οι εξισώσεις (2.9) και (2.10) αποτελούν αντίστοιχα την καταστατική εξίσωση και την εξίσωση εξόδου. Για τα γραμμικά χρονικά αμετάβλητα συστήματα με τα οποία θα ασχοληθούμε κατά κόρον στην παρούσα διπλωματική εργασία θα έχουμε την ακόλουθη γενικευμένη μορφή:

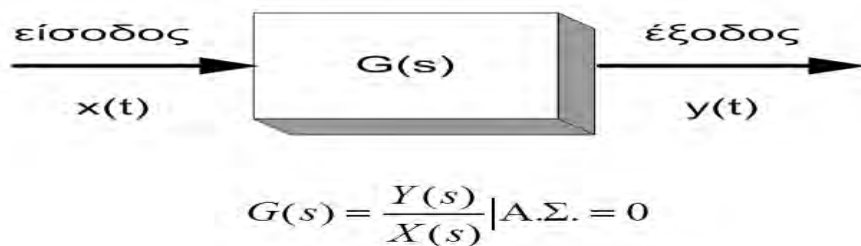
$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{b}u(t) \quad (2.11)$$

$$y(t) = \underline{c}^T \underline{x}(t) + du(t) \quad (2.12)$$

Επιπροσθέτως, τα υπό εξέταση συστήματα θα έχουν πολλαπλό αριθμό εισόδων και εξόδων (MIMO systems).

## 2.2.2 Εύρεση Συνάρτησης Μεταφοράς από Δυναμικές Εξισώσεις

Η συνάρτηση μεταφοράς χαρακτηρίζει τις σχέσεις εισόδου - εξόδου των γραμμικών χρονικά αμετάβλητων (ΓΧΑ) συστημάτων. Ως συνάρτηση μεταφοράς (transfer function) ορίζεται το πηλίκο του μετασχηματισμού Laplace της εξόδου ενός γραμμικού αμετάβλητου συστήματος προς το μετασχηματισμό Laplace της εισόδου του, όταν οι αρχικές συνθήκες είναι μηδενικές (Εικόνα 7). Η συνάρτηση μεταφοράς εξαρτάται αποκλειστικά και μόνο από τις παραμέτρους που περιγράφουν το ΓΧΑ σύστημα και είναι ανεξάρτητη από την δεδομένη είσοδο [3].



Εικόνα 7: Συνάρτηση Μεταφοράς

Σε αυτό το σημείο είμαστε πλέον σε θέση να εξάγουμε την συνάρτηση μεταφοράς ενός ΓΧΑ συστήματος, με δεδομένες τις δυναμικές εξισώσεις στον χώρο κατάστασης. Ξεκινούμε τον συλλογισμό μας από το ζεύγος δυναμικών εξισώσεων που περιγράφουν οι σχέσεις (2.9), (2.10):

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{b}u(t)$$

$$y(t) = \underline{c}^T \underline{x}(t) + du(t)$$

Παίρνοντας τον μετασχηματισμό Laplace για τις σχέσεις (2.11), (2.12) θα έχουμε αντίστοιχα:

$$s\underline{X}(s) - \underline{x}(0) = A\underline{X}(s) + \underline{b}V(s) \quad \Leftrightarrow$$

$$(sI - A) \underline{X}(s) = \underline{x}(0) + \underline{b}V(s) \quad \Leftrightarrow$$

$$\underline{X}(s) = (sI - A)^{-1}(\underline{x}(0) + \underline{b}V(s))$$

Και

$$Y(s) = \underline{c}^T \underline{X}(s) + dV(s)$$

Η συνάρτηση μεταφορά έχει νόημα για μηδενικές αρχικές συνθήκες και σε αυτήν την περίπτωση και μόνο θα έχουμε ότι:

$$\underline{X}(s) = (sI - A)^{-1} \underline{b}V(s) \Leftrightarrow \frac{\underline{X}(s)}{V(s)} = (sI - A)^{-1} \underline{b}$$

Και τελικά,

$$H(s) = \frac{Y(s)}{V(s)} = \frac{\underline{c}^T \underline{X}(s) + dV(s)}{V(s)} = \underline{c}^T (sI - A)^{-1} \underline{b} + d \quad (2.13)$$

Αξίζει να επισημάνουμε στο σημείο αυτό ότι ο βαθμός του αριθμητή θα είναι πάντα μικρότερος από τον βαθμό του παρονομαστή, ώστε να μπορεί η συνάρτηση μεταφοράς να υλοποιηθεί από ένα γραμμικό σύστημα δυναμικών εξισώσεων πεπερασμένης διάστασης. Κατά την περίπτωση, όπου ο βαθμός του πολωνύμου του αριθμητή είναι ίσος με τον βαθμό του πολωνύμου του παρονομαστή, αυτό προστίθεται από την ποσότητα  $d$ , για την οποία ισχύει  $d \neq 0$ . Τέλος, τα συστήματα με βαθμό αριθμητή μεγαλύτερο από τον αντίστοιχο βαθμό του παρονομαστή δεν είναι υλοποιήσιμα και δεν μπορούν να περιγραφούν στον χώρο κατάστασης με δυναμικές εξισώσεις πεπερασμένης διάστασης.

### 2.2.3 Ευστάθεια, Παθητικότητα ΓΧΑ Συστημάτων

Από την θεωρία των Σημάτων και Συστημάτων [3], γνωρίζουμε ότι για να είναι ένα σύστημα ευσταθές κατά ΦΕΦΕ (Φραγμένη Είσοδος – Φραγμένη Έξοδος), θα πρέπει ο βαθμός του αριθμητή να είναι μικρότερος ή ίσος από τον βαθμό του παρονομαστή. Αυτό όπως είδαμε παραπάνω αποτελεί απαίτηση για να υλοποιηθεί η συνάρτηση μεταφοράς με την βοήθεια δυναμικών εξισώσεων. Διαφορετικά, θα προκύψουν όροι της μορφής  $cs^k$ ,  $k > 0$ . Τέτοιο όροι, μπορεί να προκαλέσουν μη φραγμένα σήματα εξόδου, ακόμα και αν η είσοδος είναι φραγμένη. Για παράδειγμα, εάν ένας όρος της  $H(s)$  είναι ίσος με  $cs$  και ως είσοδο θεωρούμε την βηματική συνάρτηση  $u(t)$  με

μετασχηματισμό Laplace ίσο με  $\frac{1}{s}$ , τότε η αντίστοιχη έξοδος στο πεδίο της συχνότητας θα είναι:

$$Y(s) = H(s)V(s) = (cs) \frac{1}{s} = c$$

, η οποία προφανώς είναι μη φραγμένη.

Εναλλακτικός αλλά ισοδύναμος ορισμός είναι ότι για να είναι ένα ΓΧΑ σύστημα ευσταθές, θα πρέπει όλοι οι πόλοι της συνάρτησης μεταφοράς (ρίζες του πολυωνύμου του παρονομαστή) να έχουν αρνητικό πραγματικό μέρος, δηλαδή να βρίσκονται στο αρνητικό μιγαδικό ημιεπίπεδο. Με άλλα λόγια, οι ιδιοτιμές της χαρακτηριστικής εξίσωσης να έχουν αρνητικό πραγματικό μέρος.

Η ευστάθεια ενός συστήματος διαδραματίζει σπουδαίο ρόλο. Ειδικότερα, όταν ένας πίνακας  $A$  είναι ευσταθής, τότε με βεβαιότητα γνωρίζουμε ότι και ο  $A^{-1}$  θα είναι ευσταθής, καθώς και οι  $A^T$ ,  $A^*$ .

Παθητικά συστήματα ονομάζονται εκείνα που δεν παράγουν ενέργεια στο εσωτερικό τους [11]. Γενικά, η παθητικότητα είναι επιθυμητή ιδιότητα. Τα μη παθητικά συστήματα συνδέονται με απρόσμενες συμπεριφορές, όπως αστάθεια ή εκπομπή ενέργειας στις υψηλές συχνότητες.

## 2.2.4 Ισοδύναμα Συστήματα

Ισοδύναμα συστήματα ονομάζονται τα συστήματα, τα οποία έχουν διαφορετικές υλοποιήσεις αλλά αντιστοιχούν σε μία συνάρτηση μεταφοράς. Με άλλα λόγια έχουν αποκρίσεις συχνότητας που βρίσκονται πολύ κοντά η μία με την άλλη. Τα συστήματα αυτά έχουν διαφορετικές υλοποιήσεις αλλά όμως συσχετίζονται μεταξύ τους μέσα από μετασχηματισμούς ομοιότητας.

Πράγματι, ας θεωρήσουμε την γενική μορφή των δυναμικών συστημάτων ως εξής:

$$\dot{\underline{x}}(t) = A \underline{x}(t) + \underline{b}u(t)$$

$$y(t) = \underline{c}^T \underline{x}(t) + du(t)$$

Εάν συμβολίσουμε με  $P$  έναν αντιστρέψιμο τετραγωνικό  $n \times n$  πίνακα, με την βοήθεια μετασχηματισμών ομοιότητας θα προκύψουν οι τροποποιημένοι πίνακες:

$$A' = PAP^{-1} \quad (2.14)$$

$$\underline{b}' = P\underline{b} \quad (2.15)$$

$$\underline{c}'^T = \underline{c}^T P^{-1} \quad (2.16)$$

Και οι νέες δυναμικές εξισώσεις θα είναι:

$$\dot{\underline{x}}'(t) = \underline{A}' \underline{x}'(t) + \underline{b}' u(t) \quad (2.17)$$

$$y(t) = \underline{c}'^T \underline{x}'(t) + du(t) \quad (2.18)$$

Το τελευταίο σύστημα εξισώσεων θα έχει την ίδια συνάρτηση μεταφοράς με το αρχικό μιας και:

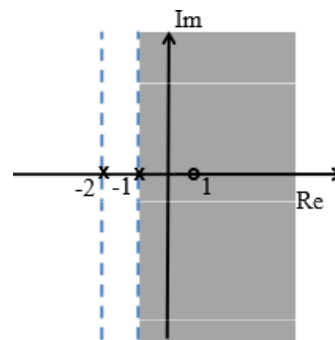
$$\begin{aligned} H'(s) &= \underline{c}'^T (s\mathbf{I} - \underline{A}')^{-1} \underline{b}' + d \\ &= \underline{c}'^T \underline{P}^{-1} (s\mathbf{I} - \underline{PAP}^{-1})^{-1} \underline{Pb} + d \\ &= \underline{c}'^T \underline{P}^{-1} [\underline{P}(s\mathbf{I} - \underline{A})\underline{P}^{-1}]^{-1} \underline{Pb} + d \\ &= \underline{c}'^T \underline{P}^{-1} \underline{P} (s\mathbf{I} - \underline{A})^{-1} \underline{P}^{-1} \underline{Pb} + d \\ &= \underline{c}'^T (s\mathbf{I} - \underline{A})^{-1} \underline{b} + d \\ &= H(s) \end{aligned} \quad (2.19)$$

## 2.2.5 Μηδενικά και Πόλοι

Κάθε σήμα  $x(t)$  που είναι γραμμικός συνδυασμός μιγαδικών εκθετικών σημάτων για  $t < 0$  και  $t > 0$  έχει ρητό μετασχηματισμό Laplace. Η αναπαράσταση του μετασχηματισμού Laplace μπορεί να γίνει στο μιγαδικό επίπεδο  $s$  ( $s$ -plane) προσδιορίζοντας τη θέση των μηδενικών και πόλων (διάγραμμα πόλων-μηδενικών/pole-zero plot).

Ορίζουμε σαν μηδενικά μιας συνάρτησης μεταφοράς, τις τιμές εκείνες για τις οποίες το πολυώνυμο του αριθμητή έχει μηδενική τιμή. Αντίστοιχα, πόλοι της συνάρτησης μεταφοράς ορίζονται οι τιμές εκείνες, για τις οποίες μηδενίζεται το πολυώνυμο του παρονομαστή της συνάρτησης.

Ένα παράδειγμα αναπαράστασης πόλων και μηδενικών φαίνεται στο διπλανό σχήμα (Εικόνα 8) όπου το -2 και το -1 είναι οι πόλοι και το 1 είναι το μηδενικό. Η περιοχή σύγκλισης καθορίζεται από τους πόλους και στο σχήμα απεικονίζεται ως η λωρίδα που είναι παράλληλη στον φανταστικό άξονα.



Εικόνα 8: Αναπαράσταση Μ/Σ Laplace στο μιγαδικό επίπεδο

Μηδενικά: ο  
Πόλοι: x

## 2.3 Ελεγχιμότητα και Παρατηρησιμότητα

### 2.3.1 Ορισμός Ελεγχιμότητας και Παρατηρησιμότητας

Παράλληλα, στο σημείο αυτό κρίνουμε απαραίτητο να ορίσουμε 2 ακόμη έννοιες που σχετίζονται με την συμπεριφορά του συστήματος, αυτές του ελέγξιμου και παρατηρήσιμου συστήματος [3]. Τα διανύσματα ελεγχιμότητας και παρατηρησιμότητας όπως θα αποδειχθεί σε μελλοντικά κεφάλαια, αποτελούν την κεντρική ιδέα με την βοήθεια της οποίας ανάγουμε το αρχικό σύστημα σε νέο, μικρότερης διάστασης.

- Η κατάσταση ενός συστήματος θα ονομάζεται ελέγξιμη (controllable) στη χρονική στιγμή  $t_0$  εάν υπάρχει πεπερασμένο  $t_1 > t_0$ , τέτοιο ώστε για κάθε  $\underline{x}(t_0)$  να υπάρχει είσοδος  $u(t)$ ,  $t \in [t_0, t_1]$ , που μπορεί να οδηγήσει την  $\underline{x}(t_0)$  σε οποιαδήποτε επιθυμητή κατάσταση  $\underline{x}(t_1)$ . Εάν αυτό εφαρμόζεται για κάθε  $t_0$ , τότε το σύστημα θα ονομάζεται ελέγξιμο (controllable). Με άλλα λόγια, για να είμαστε σε θέση να εκμεταλλευτούμε το σύστημα και να οδηγήσουμε την έξοδο όπου εμείς θέλουμε, θα πρέπει να έχουμε την δυνατότητα να επιλέγουμε τις αρχικές συνθήκες και να οδηγούμε τις μεταβλητές κατάστασης στις τιμές που επιθυμούμε, εφαρμόζοντας την κατάλληλη είσοδο.
- Η κατάσταση του συστήματος την χρονική στιγμή  $t = t_0$  θα λέμε ότι είναι παρατηρήσιμη, εάν υπάρχει πεπερασμένο  $t_1$ , έτσι ώστε αν γνωρίζουμε την είσοδο στο διάστημα  $[t_0, t_1]$  και την αντίστοιχη έξοδο στο ίδιο διάστημα, να μπορούμε να υπολογίσουμε την κατάσταση  $\underline{x}(t_0)$ . Εάν αυτό εφαρμόζεται για κάθε  $t_0$ , τότε το σύστημα θα ονομάζεται παρατηρήσιμο (Observable). Πιο συγκεκριμένα, η έννοια της παρατηρησιμότητας αναφέρεται στην δυνατότητα να μπορούμε να υπολογίζουμε τις τιμές που έχουν οι μεταβλητές κατάστασης οποιαδήποτε χρονική στιγμή, γνωρίζοντας μόνο τα σήματα εισόδου και εξόδου.

### 2.3.2 Ταυτόχρονη Ελεγχιμότητα και Παρατηρησιμότητα

Ένα γραμμικό, χρονικά αμετάβλητο σύστημα θα είναι ταυτόχρονα ελέγξιμο και παρατηρήσιμο, εάν η συνάρτηση μεταφοράς του είναι μη - συρρικνούμενη, δηλαδή δεν υπάρχουν κοινοί πόλοι και μηδενικά στον παρονομαστή και στον αριθμητή αντίστοιχα. Με τον όρο μη - συρρικνούμενη εννοούμε ότι έχουμε λάβει την ελάχιστη υλοποίηση του συστήματος. Είναι προφανές ότι εάν η συνάρτηση μεταφοράς είναι σε συρρικνούμενη μορφή, τότε απαλείφοντας σταδιακά κοινούς πόλους και κοινά μηδενικά θα καταλήξουμε σε δυναμικές εξισώσεις με συνεχώς ελαττωμένη διάσταση με αποτέλεσμα να πάρουμε ένα μαθηματικό μοντέλο μικρότερης τάξης. Όλες αυτές οι δυναμικές εξισώσεις που προκύπτουν θα αντιστοιχούν στην ίδια συνάρτηση μεταφοράς.

## 2.4 Modified Nodal Analysis (MNA) Εξισώσεις

Η τροποποιημένη ανάλυση κόμβων (Modified Nodal Analysis - MNA) είναι μία προέκταση της κλασσικής μεθόδου κόμβων (Nodal Analysis - NA), η οποία αντιμετωπίζει το κύκλωμα σαν αυτό να μην περιέχει καθόλου πηγές τάσης και μας δίνει την δυνατότητα να υπολογίζουμε τα ρεύματα κατά μήκος μερικών κλάδων [4]. Χρησιμοποιεί τους νόμους του Kirchhoff καθώς και τις χαρακτηριστικές εξισώσεις ρευμάτων και τάσεων. Για να εξάγουμε τις εξισώσεις του κυκλώματος, πρέπει να λάβουμε υπόψιν την τοπολογία του κυκλώματος. Η MNA ανάλυση είναι εύκολο να μοντελοποιηθεί και εφαρμόζεται εξίσου καλά σε αναλύσεις τόσο στο πεδίο του χρόνου όσο και στο πεδίο της συχνότητας.

### Εξισώσεις Τοπολογίας

Έστω  $V = 0, 1, \dots, n - 1$  το σύνολο των  $n$  κόμβων (nodes) ή κορυφών (vertices) και  $E = e_1, e_2, \dots, e_m$  το σύνολο των  $m$  κλάδων (branches) ή ακμών (edges) του κυκλώματος, όπου ο κόμβος αναφοράς (γείωση) συμβολίζεται με 0. Ο ελαττωμένος πίνακας πρόσπτωσης  $A \in R^{(n-1) \times m}$ , του γράφου  $G(V, E)$  του κυκλώματος ορίζεται ως εξής:

$$A_{i,j} = \begin{cases} +1, & \text{εαν ο κλάδος } j \text{ εξέρχεται από τον κόμβο } i \\ -1, & \text{εαν ο κλάδος } j \text{ εισέρχεται στον κόμβο } i \\ 0, & \text{εαν ο κλάδος } j \text{ δε συνδέεται με τον κόμβο } i \end{cases}$$

Οι διαστάσεις του πίνακα  $A$  είναι  $(n - 1) \times m$ .

Στα πλαίσια αυτής της έρευνας, δεν κρίνουμε αναγκαίο να εστιάσουμε στην απόδειξη του τρόπου παραγωγής των MNA εξισώσεων μέσω νόμων ρευμάτων και κόμβων, μιας και η σκοπιμότητα της εργασίας δεν είναι αυτή. Μετά τη θεωρητική ανάλυση που θα πραγματοποιηθεί ακολούθως για τη δημιουργία του πίνακα, εμείς δεν χρειάζεται να ασχοληθούμε ξανά με τους νόμους Kirchhoff.

Ενδεικτικά και προσπαθώντας να διατηρήσουμε την περιγραφή όσο γίνεται απλούστερη αναφέρουμε συνοπτικά τα 4 βήματα που λαμβάνουν χώρα για την εξαγωγή των πινάκων  $C, G$  με την βοήθεια εξισώσεων:

### **Βήμα 1**

Γράψε τους Νόμους Ρευμάτων Kirchhoff για το σύστημα στη μορφή:  $Ai = 0$ , όπου  $A$  ο πίνακας συχνότητας για τα ρεύματα σε κάθε κόμβο του κυκλώματος. Αξιοσημείωτο είναι το γεγονός ότι στις ανεξάρτητες πηγές τάσης το ρεύμα λαμβάνεται ως θετικό όταν έχει φορά από το θετικό προς τον αρνητικό πόλο και αρνητικό σε αντίθετη περίπτωση. Στους κλάδους, τα ρεύματα που εισέρχονται σε έναν κόμβο έχουν αρνητικό πρόσημο ενώ όσα εξέρχονται θετικό.



**Βήμα 2**

Χρησιμοποιώντας τις εξισώσεις των στοιχείων απάλειψε όσες περισσότερες μεταβλητές ρευμάτων από τους Νόμους Ρευμάτων Kirchhoff.

**Βήμα 3**

Χρησιμοποιώντας τους Νόμους Τάσεων Kirchhoff αντικατάστησε τις τάσεις των κλάδων με τα δυναμικά των κόμβων του κυκλώματος.

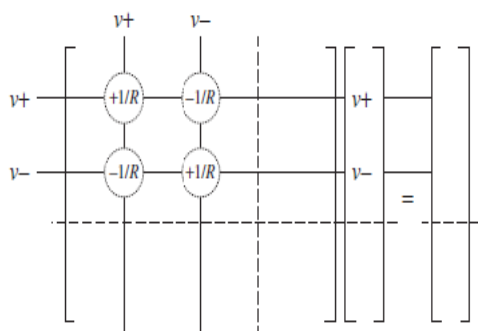
**Βήμα 4**

Συγκέντρωσε τις εξισώσεις των στοιχείων εκείνων για τα οποία οι μεταβλητές ρεύματος δεν μπόρεσαν να επαλειφθούν ως επιπλέον εξισώσεις για το σύστημα.

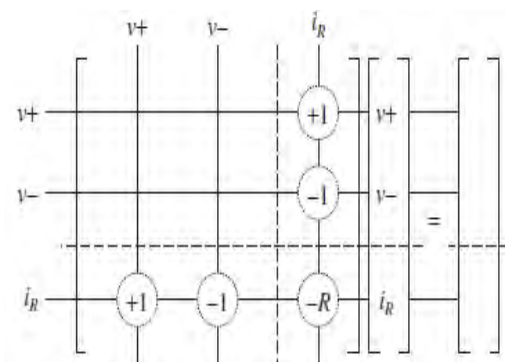
Στην πράξη, για την κατασκευή ενός προσομοιωτή, οι παραπάνω πίνακες εξισώσεων δεν χρησιμοποιούνται για την διαμόρφωση του MNA συστήματος. Για κάθε κυκλωματικό στοιχείο υπάρχει μια “συνεισφορά” στον πίνακα εξισώσεων η οποία ονομάζεται “σφραγίδα” του στοιχείου και η οποία μας λέει ακριβώς σε ποιο σημείο και με τι βάρος θα συνεισφέρει στη δομή που θα αντιπροσωπεύει τον MNA πίνακα. Με βάση αυτήν την συνεισφορά και όπως θα δούμε αναλυτικότερα παρακάτω διαμορφώνεται το MNA σύστημα, σχεδόν άμεσα τη στιγμή που διαβάζουμε το στοιχείο από το αρχείο εισόδου.

**Αντιστάτες:**

Θεωρούμε αντιστάτη<sup>2</sup> με τους κόμβους των ακροδεκτών του να συμβολίζονται με  $\mathbf{n}+$  και  $\mathbf{n}-$ , καθώς επίσης και τα δυναμικά σε αυτούς με  $\mathbf{v}+$  και  $\mathbf{v}-$  αντίστοιχα. Το ρεύμα του αντιστάτη που συμβολίζεται με  $\mathbf{i}_R$  έχει κατεύθυνση από το  $\mathbf{n}+$  προς τον  $\mathbf{n}-$ . Ανάλογα με την ομάδα στην οποία ο αντιστάτης ανήκει, οι σφραγίδες για τη συνεισφορά του στοιχείου αυτού στο σύστημα φαίνονται παρακάτω στις εικόνες 9 και 10:



Εικόνα 9: Σφραγίδες αντιστατών πρώτης ομάδας

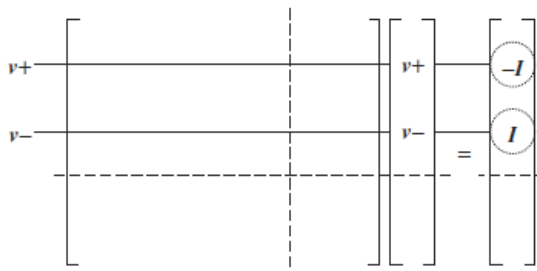


Εικόνα 10: Σφραγίδες αντιστατών δεύτερης ομάδας

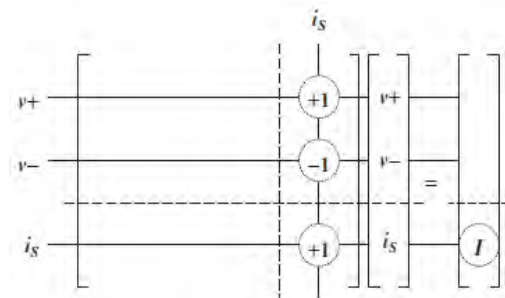
<sup>2</sup> Ο **αντιστάτης** είναι ένα ηλεκτρολογικό/ηλεκτρονικό εξάρτημα το οποίο χρησιμοποιείται σε διάφορα κυκλώματα για τον έλεγχο της ροής του ρεύματος.

Ανεξάρτητες πηγές ρεύματος:

Θεωρούμε ανεξάρτητη πηγή ρεύματος με χαρακτηριστική εξίσωση  $i_S = I$  και με κατεύθυνση αναφοράς από τον κόμβο  $n+$  προς τον  $n-$  οι οποίοι έχουν δυναμικά  $v+$  και  $v-$  αντίστοιχα. Ανάλογα με την ομάδα στην οποία ανήκει η ανεξάρτητη πηγή ρεύματος, οι σφραγίδες για τη συνεισφορά του στοιχείου αυτού στο σύστημα φαίνονται παρακάτω στις εικόνες 11 και 12:



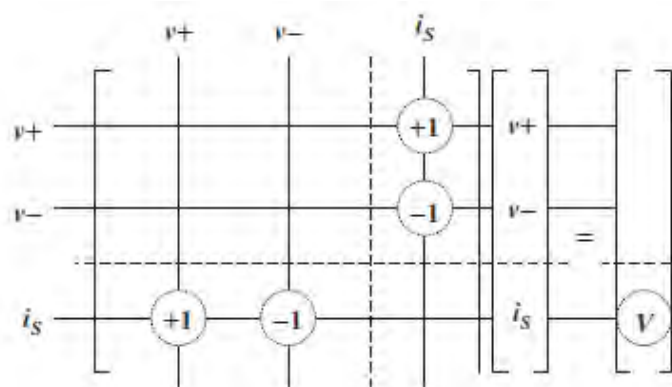
Εικόνα 11: Σφραγίδες πηγών ρεύματος πρώτης ομάδας



Εικόνα 12: Σφραγίδες πηγών ρεύματος δεύτερης ομάδας

Ανεξάρτητες πηγές τάσης:

Θεωρούμε ανεξάρτητη πηγή τάσης με χαρακτηριστική εξίσωση  $v+ - v- = V$  και με κατεύθυνση αναφοράς από τον κόμβο  $n+$  προς τον  $n-$  οι οποίοι έχουν δυναμικά  $v+$  και  $v-$  αντίστοιχα. Οι σφραγίδες για τη συνεισφορά του στοιχείου αυτού στο σύστημα φαίνονται παρακάτω στην εικόνα 13:



Εικόνα 13: Σφραγίδες πηγών τάσης

Τα βήματα συνοπτικά για παραγωγή των MNA εξισώσεων με επισκόπηση της τοπολογίας του κυκλώματος είναι ως εξής:

### **Βήμα 1**

Ο πίνακας αρχικοποιείται με μηδενικά.

### **Βήμα 2**

Το διάνυσμα του δεξί μέλους (RHS) αρχικοποιείται με μηδέν.

### **Βήμα 3**

Οι σφραγίδες των στοιχείων προστίθενται στον πίνακα και το διάνυσμα του δεξί μέλους συμπληρώνεται με τα ονόματα των στοιχείων.

## **2.5 Συσχετισμός των MNA εξισώσεων με τις Δυναμικές Εξισώσεις**

Ας θεωρήσουμε ένα γραμμικά χρονικά αμετάβλητο (ΓΧΑ) σύστημα με διάνυσμα πολλαπλών εισόδων και εξόδων (Multi Input - Multi Output, MIMO) σε MNA μορφή, όπου  $n$  είναι ο συνολικός αριθμός των μεταβλητών κατάστασης και  $N_i$ ,  $N_o$ , ο αριθμός των εισόδων και εξόδων αντίστοιχα.

$$G\underline{x}(t) + C\dot{\underline{x}}(t) = B\underline{u}_{in}(t) \quad (2.20)$$

$$\underline{V}_{out}(t) = L^T \underline{x}(t) \quad (2.21)$$

- Οι πίνακες  $G$ ,  $C$  αποτελούν τετραγωνικούς  $n \times n$  πίνακες, οι οποίοι προκύπτουν από τα στοιχεία και την τοπολογία του κυκλώματος
- Το  $\underline{x}(t)$  είναι το διάνυσμα των μεταβλητών κατάστασης με μέγεθος  $n$
- $\underline{u}_{in}(t)$  αποτελεί το διάνυσμα των εισόδων με μέγεθος  $N_i$
- $\underline{V}_{out}(t)$  αποτελεί το διάνυσμα των εξόδων με μέγεθος  $N_o$
- $B$  είναι ο πίνακας  $n \times N_i$ , που αντιστοιχίζει το διάνυσμα των εισόδων  $\underline{u}_{in}(t)$  στο διάνυσμα των μεταβλητών κατάστασης  $\underline{x}(t)$
- $L$  είναι ο πίνακας  $n \times N_o$  που αντιστοιχίζει το διάνυσμα των εξόδων  $\underline{V}_{out}(t)$  στο διάνυσμα των μεταβλητών κατάστασης  $\underline{x}(t)$

Σε παραπάνω ενότητα δείξαμε τον τρόπο υπολογισμού της συνάρτησης μεταφοράς ενός συστήματος που περιγράφεται με την βοήθεια δυναμικών εξισώσεων, κάνοντας χρήση των μετασχηματισμών Laplace. Πράττοντας όμοια και στις εξισώσεις της (2.20), (2.21) καταλήγουμε στην εξής συνάρτηση μεταφοράς:

$$H(s) = L^T (G + sC)^{-1}B \quad (2.22)$$

Θα αποδείξουμε λοιπόν πως οι παραπάνω MNA εξισώσεις (2.20), (2.21) είναι όμοιες με τις δυναμικές εξισώσεις και συνεπώς η συνάρτηση μεταφοράς τους περιγράφεται από το ίδιο μαθηματικό μοντέλο που περιγράφεται και η συνάρτηση μεταφοράς των δυναμικών εξισώσεων [2]. Υποθέτουμε πως ο πίνακας C είναι αντιστρέψιμος, δηλαδή υπάρχει και ορίζεται ο πίνακας  $C^{-1}$  και έτσι έχουμε:

$$\begin{aligned} G \underline{x}(t) + C \dot{\underline{x}}(t) &= B u_{in}(t) && \Leftrightarrow \\ C^{-1}G \underline{x}(t) + C^{-1}C \dot{\underline{x}}(t) &= C^{-1}B u_{in}(t) && \Leftrightarrow \\ \dot{\underline{x}}(t) &= -C^{-1}G \underline{x}(t) + C^{-1}B u_{in}(t) && (2.23) \end{aligned}$$

Αν θέσουμε,  $A = -C^{-1}G$  και  $\underline{b} = C^{-1}B$  στην σχέση (2.23) θα προκύψει ότι:

$$\dot{\underline{x}}(t) = A \underline{x}(t) + \underline{b} u_{in}(t) \quad (2.24)$$

Συγχρόνως, η έξοδος είναι ίδια με πριν, με μοναδική διαφορά ότι τώρα ισχύει  $c = L$ , δηλαδή:

$$V_{out}(t) = L^T \underline{x}(t) \quad (2.25)$$

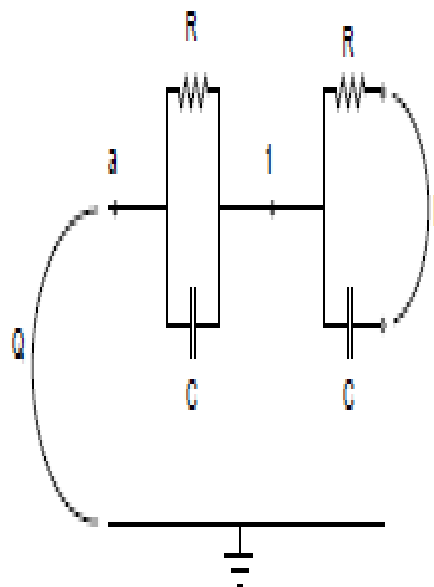
οι οποίες αποτελούν την καταστατική εξίσωση και την εξίσωση εξόδου αντίστοιχα στη μορφή που δείξαμε στη σχέση (2.11) και (2.12) παραπάνω. Έτσι λοιπόν, η συνάρτησης μεταφοράς των MNA εξισώσεων θα ταυτίζεται με την συνάρτηση μεταφοράς των δυναμικών εξισώσεων και συγκεκριμένα θα είναι:

$$H(s) = \underline{c}^T (sI - A)^{-1} \underline{b} + \underline{d}$$

### 2.5.1 Τερματικά και Θύρες

Τερματικό είναι ένας κόμβος του κυκλώματος ο οποίος είναι προσπελάσιμος από το εξωτερικό του κυκλώματος, δηλαδή είναι ένας κόμβος στον οποίο μπορεί να “εγχυθεί” ρεύμα (κόμβος  $a$  στην εικόνα 14). Συχνά το τερματικό χαρακτηρίζεται και ως εξωτερικός κόμβος σε σύγκριση με τον χαρακτηρισμό του εσωτερικού κόμβου ο οποίος δεν είναι προσπελάσιμος από το εξωτερικό του κυκλώματος (όπως ο κόμβος 1 στην εικόνα 14). Μία θύρα αποτελείται από δύο τερματικά τα οποία μπορούν να συνδεθούν μέσω μιας πηγής ρεύματος ή τάσης ή μέσω ενός άλλου υπό κυκλώματος (όπως η θύρα  $P$  στην εικόνα 14). Συνήθως θεωρείται ότι το τερματικό αναφοράς μιας θύρας είναι η γείωση. Έτσι, το τερματικό  $a$  (στην εικόνα 14) συμπεριλαμβανομένης και της γείωσης, μπορεί να θεωρηθεί ως θύρα ( $Q$ ). Στο πλαίσιο της παρούσης εργασίας όταν αναφερόμαστε σε μια θύρα, έστω  $X$ , θα

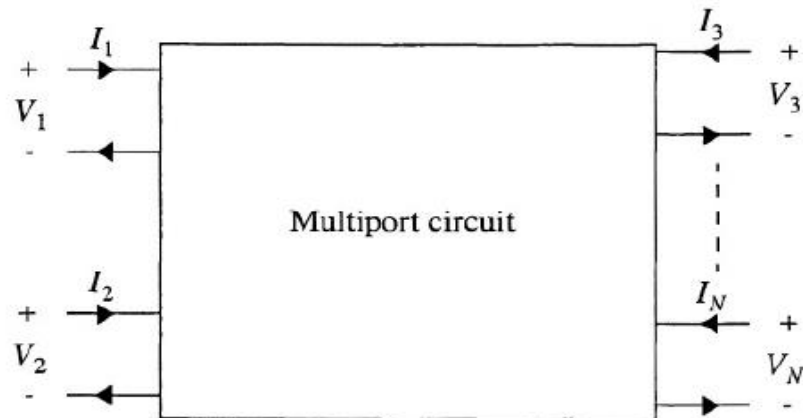
αναφερόμαστε στην θύρα που αποτελείται από το αντίστοιχο τερματικό  $X$  και την γείωση.



Εικόνα 14: Κύκλωμα με τερματικό  $a$ , εσωτερικό κόμβο 1, θύρα  $P$  και θύρα  $Q(a,0)$

Κάθε γραμμικό κύκλωμα, μπορεί να χαρακτηριστεί πλήρως από τον πίνακα συναρτήσεων μεταφοράς στο πεδίο της μιγαδικής συχνότητας  $s$ . Σημειώνουμε πως γνωρίζοντας τη συνάρτηση μεταφοράς  $T(s)$  μπορούμε να την μελετήσουμε για φυσικές συχνότητες με αντικατάσταση του  $s$  με  $j\omega$ . Η συνάρτηση μεταφοράς είναι μια σύνθετη ποσότητα όπου το μέτρο της δίνει την απόκριση μέτρου, ενώ η γωνία της την απόκριση φάσης.

Οι  $Y$  και  $Z$  παράμετροι είναι υποσύνολα της γενικότερης κατηγορίας των πινάκων συναρτήσεων μεταφοράς. Ανάλογα με την συμπεριφορά κάθε θύρας εισόδου και εξόδου η συνάρτηση μεταφοράς μπορεί να περιγράφεται είτε από την σχέση  $YV = I$  είτε από την σχέση  $ZI = V$  [2]. Οι πίνακες  $Y$ ,  $Z$  περιγράφουν ουσιαστικά την συμπεριφορά ενός ηλεκτρικού κυκλώματος, το οποίο για τους σκοπούς της περιγραφής μπορούμε να το σκεφτούμε ως ένα "μαύρο κουτί" με  $N$  θύρες εισόδου και  $N$  θύρες εξόδου (Εικόνα 15).

Εικόνα 15: Κύκλωμα με  $N$  θύρες

Στην πρώτη περίπτωση του πίνακα  $Y$  των αγωγιμοτήτων, διάστασης  $n \times n$ , λαμβάνουμε υπόψιν ως διάνυσμα εισόδου τις πηγές τάσης και ως διάνυσμα εξόδου τα ρεύματα κατά μήκος αυτών των πηγών τάσης (admittance). Στις εισόδους έχουμε πηγές τάσης και άρα αντικαθιστούμε το διάνυσμα εισόδου  $u_{in}(t)$  με  $u_p(t)$  και το διάνυσμα  $u_{out}(t)$  της εξόδου που αντιστοιχεί σε ρεύματα, με  $i_p(t)$ . Συνεπώς, οι MNA εξισώσεις μετατρέπονται ως εξής:

$$G\underline{x}(t) + C\dot{\underline{x}}(t) = B u_p(t)$$

$$i_p(t) = -B^T \underline{x}(t), \quad \text{όπου } L = -B$$

Ο αντίστοιχος πίνακας αγωγιμοτήτων θα είναι:

$$Y(s) = -B^T (G + sC)^{-1} B \quad (2.26)$$

Αντιθέτως, στην δεύτερη περίπτωση του πίνακα  $Z$ , θεωρούμε ως είσοδο το διάνυσμα των πηγών ρεύματος και παίρνουμε την έξοδο μετρώντας την τάση κατά μήκος αυτών των πηγών ρεύματος (impedance). Στην μορφή  $Z$  αντικαθιστούμε το  $u_{in}(t)$  με  $i_p(t)$ , αφού ως είσοδο λαμβάνουμε τις πηγές ρεύματος και το  $u_{out}(t)$  με  $u_p(t)$ .

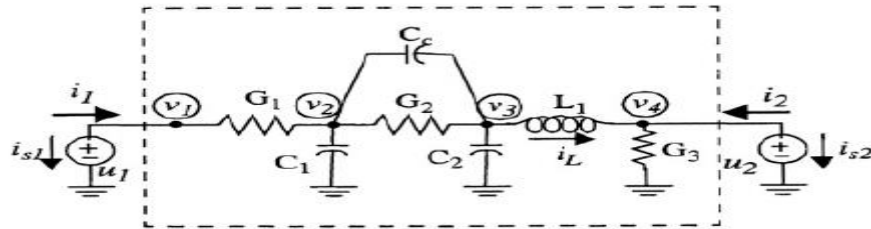
$$G\underline{x}(t) + C\dot{\underline{x}}(t) = B i_p(t)$$

$$u_p(t) = B^T \underline{x}(t), \quad \text{όπου } L = B$$

Ο αντίστοιχος πίνακας θα είναι:

$$Z(s) = B^T (G + sC)^{-1} B \quad (2.27)$$

Στο σημείο αυτό για να γίνουν περισσότερο κατανοητά όσα περιγράψαμε παραπάνω, ας παραθέσουμε ένα κύκλωμα με 2 θύρες (Εικόνα 16) και τους αντίστοιχους MNA πίνακες στην μορφή Y που προκύπτουν από την τοπολογία του κυκλώματος.



Εικόνα 16: Κύκλωμα για εξαγωγή MNA πινάκων σε Y μορφή

$$G = \begin{bmatrix} G1 & -G1 & 0 & 0 & 0 & 1 & 0 \\ -G1 & (G1+G2) & -G2 & 0 & 0 & 0 & 0 \\ 0 & -G2 & G2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & G3 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & Cc + C1 & -Cc & 0 & 0 & 0 & 0 \\ 0 & -Cc & Cc + C2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -L1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{\chi} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ I_L \\ I_{S1} \\ I_{S2} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$U_p = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad I_p = \begin{bmatrix} i_1 \\ i_2 \end{bmatrix}, \quad L = -B$$

## 2.6 Έννοιες Γραμμικής Άλγεβρας

Στην συνέχεια, παραθέτουμε μερικούς ορισμούς από την θεωρία πινάκων της γραμμικής άλγεβρας, οι οποίοι είναι απαραίτητοι για την κατανόηση της περιγραφής των βημάτων του αλγορίθμου στο κεφάλαιο 5.

### 1) Συμμετρικός Πίνακας

- a) Συμμετρικός<sup>3</sup> πίνακας ονομάζεται κάθε τετραγωνικός πίνακας που είναι ίσος με τον ανάστροφο του [12]. Τα στοιχεία του πίνακα είναι συμμετρικά ως προς την κύρια διαγώνιο. Μαθηματικά ορισμένο είναι:

$$A = A^T$$

- b) Εάν ο  $A^{-1}$  ορίζεται, θα είναι συμμετρικός, αν και μόνο αν ο  $A$  είναι συμμετρικός.

### 2) Θετικά ορισμένος Πίνακας

- a) Ένας πραγματικός, συμμετρικός  $n \times n$  πίνακας  $A$  θα είναι θετικά ορισμένος (positive definitive) αν και μόνο αν το βαθμωτό γινόμενο  $z^T A z$  είναι θετικό, για κάθε μη μηδενική στήλη του διανύσματος  $z$ , η οποία αποτελείται από  $n$  πραγματικές εγγραφές [13].
- b) Κατά την παραγοντοποίηση Cholesky, ο πίνακας  $A$  θα είναι θετικά ορισμένος, αν και μόνο αν υπάρχει ένας μοναδικός κάτω τριγωνικός πίνακας  $L$ , ο οποίος έχει πραγματικές και αυστηρά θετικές τιμές στην διαγώνιό του, έτσι ώστε  $A = LL^*$ , όπου  $L^*$  ανάστροφος πίνακας του  $L$ .
- c) Για κάθε αντιστρέψιμο πίνακα  $A$  ( $\exists A^{-1}$ ), το γινόμενο  $A^T A$  θα είναι ένας θετικά ορισμένος πίνακας.

### 3) Μετασχηματισμός Ομοιότητας

---

<sup>3</sup> Κάθε τετραγωνικός, διαγώνιος πίνακας είναι συμμετρικός, αφού τα στοιχεία πάνω και κάτω από την κύρια διαγώνιο είναι μηδέν.



- a) Στον μετασχηματισμό ομοιότητας, ένας πίνακας  $A$  μπορεί να διαγωνοποιηθεί ως εξής:  $A = P\Lambda P^{-1}$ , όπου ο  $\Lambda$  είναι ένας διαγώνιος πίνακας και περιέχει τις ιδιοτιμές του  $A$  ενώ ο  $P$  αποτελείται από τα αντίστοιχα ιδιοδιανύσματα.
- b) Δύο πραγματικοί, συμμετρικοί, θετικά ορισμένοι πίνακες  $A, B$  διάστασης  $n \times n$  θα είναι συγχρόνως διαγωνοποιήσιμοι, εάν και μόνο αν υπάρχει μια βάση, έτσι ώστε κάθε στοιχείο της βάσης να αποτελεί ιδιοδιάνυσμα και για τον  $A$  και για τον  $B$ . Επιτυγχάνεται μέσω ενός μετασχηματισμού ομοιότητας.

#### 4) Ορθοκανονική Βάση

- a) Ορθογώνια βάση ενός διανυσματικού χώρου είναι μια βάση στην οποία κάθε διάνυσμά της είναι κάθετο σε κάθε άλλο διάνυσμα της βάσης.
- b) Ορθοκανονική βάση ενός διανυσματικού χώρου ονομάζουμε μια ορθογώνια βάση, της οποίας κάθε διάνυσμα έχει μέτρο 1.

#### 5) Ιδιοτιμές/Ιδιοδιανύσματα

Έστω ένας τετραγωνικός πίνακας τάξης  $n$ . Η τιμή  $\lambda \in \mathbb{R}$  θα λέγεται ιδιοτιμή του  $A$  αν υπάρχει ένα μη μηδενικό διάνυσμα-στήλη  $u \in \mathbb{R}^n$  τέτοιο ώστε:

$$Au = \lambda u$$

Κάθε διάνυσμα που ικανοποιεί τη σχέση αυτή ονομάζεται ιδιοδιάνυσμα του  $A$  που αντιστοιχεί στην ιδιοτιμή  $\lambda$ . Επίσης ένας πίνακας μπορεί να έχει πολλές ιδιοτιμές και σε κάθε ιδιοτιμή να αντιστοιχούν πολλά ιδιοδιανύσματα.

#### Εύρεση ιδιοτιμής

- Λύνουμε την εξίσωση  $\det(A - \lambda I) = 0$ .
- Στην συνέχεια, για κάθε ιδιοτιμή  $\lambda$  ξεχωριστά λύνουμε το ομογενές σύστημα:

$$(A - \lambda I)X = 0$$

, ώστε να βρούμε τον αντίστοιχο ιδιοχώρο.

Η σελίδα αυτή είναι σκόπιμα λευκή.

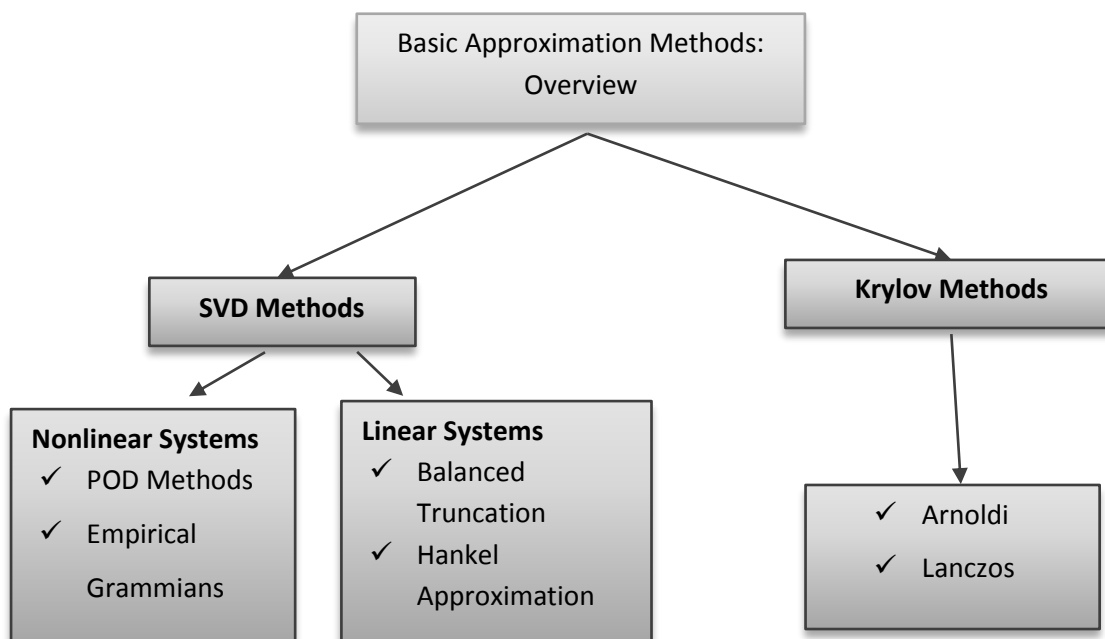
# ΚΕΦΑΛΑΙΟ 3

## Μέθοδοι Υποβιβασμού Τάξης Μοντέλου

Όπως έχουμε ήδη επισημάνει, οι αλγόριθμοι μείωσης τάξης μοντέλου διευκολύνουν την ανάλυση, την προσομοίωση και τον σχεδιασμό των συστημάτων αυτομάτου ελέγχου. Μια ευρεία γκάμα αλγορίθμων συναντάται στην επιστημονική βιβλιογραφία για γραμμικά και μη γραμμικά συστήματα τόσο στο πεδίο του χρόνου όσο και στο πεδίο της συχνότητας και εξετάζονται στο παρόν κεφάλαιο. Επίσης, παρουσιάζονται τα ιδιαίτερα χαρακτηριστικά καθώς και η φυσική ερμηνεία του αλγορίθμου TBR που επιλέχθηκε προς εξέταση.

### 3.1 Επισκόπηση μεθόδων MOR

Οι αλγόριθμοι μείωσης τάξης μοντέλου διακρίνονται σε δύο μεγάλες κατηγορίες, στις τεχνικές Singular Value Decomposition (SVD) και στις τεχνικές Moment Matching (μέθοδοι Krylov). Καθένας από τους αλγόριθμους που εντάσσεται στην κάθε κατηγορία παρουσιάζει, ανάλογα με το είδος της εφαρμογής που θα τον αξιοποιήσει, τα ισχυρά και τα αδύνατα σημεία του [9]. Βέβαια, ένας αλγόριθμος που δίνει καλές προσεγγίσεις σε ένα συγκεκριμένο σύστημα δεν συνεπάγεται ότι θα αποδίδει εξίσου καλά σε οποιοδήποτε σύστημα. Ο σκοπός αυτού του κεφαλαίου της διπλωματικής εργασίας, είναι να περιγράψουμε συνοπτικά την κεντρική ιδέα με την οποία οδηγούμαστε στο μοντέλο μικρότερης κλίμακας για μερικούς από τους πιο διαδεδομένους αλγορίθμους των δύο αυτών μεγάλων κατηγοριών (Σχήμα 1). Συγχρόνως, θα αναφερθούμε στις εγγυήσεις που προσφέρει ο κάθε αλγόριθμος, καθώς και τα σκοτεινά του σημεία. Ο λόγος που επιλέγουμε να κάνουμε αυτήν την επισκόπηση στην επιστημονική βιβλιογραφία των MOR μεθόδων, είναι για να γίνει εμφανής η επιλογή ενασχόλησης με τον αλγόριθμο TBR.



Σχήμα 1: Συνοπτική παρουσίαση μεθόδων MOR

Αρχικά, αναφέρουμε ότι οι αλγόριθμοι της κατηγορίας SVD, αφενός παρέχουν υπολογίσιμα όρια σφάλματος και διατήρηση των ιδιοτήτων του αρχικού συστήματος (όπως παθητικότητα, ευστάθεια), αφετέρου όμως δεν είναι υπολογιστικά αποδοτικοί για μεγάλης κλίμακας συστήματα με χρονική και χωρική πολυπλοκότητα  $O(n^3)$  και  $O(n^2)$  αντίστοιχα, όπου  $n$  η διάσταση του συστήματος [8]. Στον αντίποδα, οι μέθοδοι Krylov, είναι ανεκτοί υπολογιστικά με πολυπλοκότητα  $O(r^2n)$  όπου  $n$  ο αριθμός των καταστάσεων, αλλά όμως δεν δίνουν καμία εγγύηση για συνέχιση κατοχής των ιδιοτήτων του συστήματος ούτε και για υπολογίσιμα όρια λάθους. Συνεπώς, αναλόγως με το είδος της εφαρμογής και τις προδιαγραφές στις οποίες θέλουμε να ανταποκριθούμε επιλέγουμε και έναν αλγόριθμο της αντίστοιχης κατηγορίας.

### SVD Μέθοδοι

Στην κατηγορία SVD μεθόδων εντοπίζουμε αλγόριθμους κατάλληλους τόσο για γραμμικά όσο και για μη γραμμικά δυναμικά συστήματα. Όσον αφορά στις εφαρμογές των μη γραμμικών δυναμικών συστημάτων, ο αλγόριθμος Proper Orthogonal Decomposition (POD) κατέχει περίοπτη θέση [9]. Ο συγκεκριμένος αλγόριθμος χρησιμοποιείται ευρέως στην μείωση μεγεθών συστημάτων στην επεξεργασία εικόνας, στην συμπίεση δεδομένων καθώς και στην ανάλυση σημάτων. Η βασική ιδέα του έγκειται στο να μειώσει τον αρχικά μεγάλο αριθμό εξαρτημένων μεταβλητών. Μετασχηματισμοί ορθογωνιότητας εφαρμόζονται στα ιδιοδιανύσματα του πίνακα συνδιασποράς. Στην συνέχεια, τα δεδομένα προβάλλονται στον υποχώρο που σχηματίζεται από τα ιδιοδιανύσματα που αντιστοιχούν στις μεγαλύτερες ιδιοτιμές. Η πιο αξιοσημείωτη ιδιότητα του είναι ότι ελαχιστοποιεί την μέση τετραγωνική

απόσταση μεταξύ του αρχικού και του μειωμένου συστήματος. Η εξαρτημένη φύση του σήματος της POD μπορεί να θεωρηθεί ως ένα από τα πιο αδύναμα σημεία της μεθόδου.

### Krylov Μέθοδοι

Ο υποχώρος Krylov τάξης  $r$ , παράγεται από έναν πίνακα  $A$  διάστασης  $n \times n$  και ένα διάνυσμα  $b$  διάστασης  $n$  με τον εξής τρόπο:

$$K_r(A,b) = \text{span} \{b, Ab, A^2b, \dots, A^{r-1}b\} \quad (3.1)$$

Χρησιμοποιείται για την επίλυση μεγάλης τάξης συστημάτων ή γραμμικών εξισώσεων, ώστε να αποφύγουμε τις δαπανηρές πράξεις πίνακα με πίνακα. Πολλαπλασιάζουμε αρχικά, τον πίνακα  $A$  με το διάνυσμα  $b$  ( $Ab$ ), και το αποτέλεσμα που είναι επίσης διάνυσμα το πολλαπλασιάζουμε ξανά με τον  $A$ , ώστε να πάρουμε το διάνυσμα:  $Ab * A = A^2b$ . Η διαδικασία συνεχίζεται ανάλογα μέχρις ότου να φτάσουμε στο διάνυσμα  $A^{r-1}b$ . Κάθε φορά έχουμε να αντιμετωπίσουμε τον πολλαπλασιασμό πίνακα με διάνυσμα και έτσι οι μέθοδοι υποβιβασμού τάξης που βασίζονται στους υποχώρους Krylov γίνονται πολύ αποδοτικοί.

Τα μικρότερης τάξης μοντέλα παράγονται μέσω μιας διαδικασίας προβολής.

Δηλαδή:

$$\tilde{A} = W^T A V \quad (3.2)$$

$$\tilde{B} = W^T B \quad (3.3)$$

$$\tilde{C} = C V \quad (3.4)$$

$$\tilde{D} = D \quad (3.5)$$

Οι πίνακες προβολής, έστω  $V, W$ , κατασκευάζονται έτσι ώστε οι στήλες τους να είναι τα διανύσματα που προκύπτουν από κάποιον υποχώρο Krylov. Για παράδειγμα, ο αλγόριθμος Arnoldi κατασκευάζει τους πίνακες προβολής έτσι ώστε  $V=W$ , ενώ ο αλγόριθμος Lanczos φτιάχνει δύο υποχώρους Krylov για τον σχηματισμό των πινάκων  $V, W$ .

## **3.2 Ιδιαίτερα Χαρακτηριστικά της μεθόδου TBR**

Η μέθοδος που έχει επιλεγεί για υλοποίηση, προσομοίωση και μελέτη των χαρακτηριστικών της απόκρισης της στα πλαίσια αυτής της διπλωματικής εργασίας είναι η Truncated Balanced Realization (Εικόνα 17). Οι βασικοί λόγοι που ξεχωρίζει μεταξύ των υπολοίπων Model Order Reduction μεθόδων, είναι ότι:

- προσφέρει την δυνατότητα για υπολογίσιμα όρια σφάλματος

- διατηρεί αναλλοίωτες τις ιδιότητες του αρχικού συστήματος, όπως παθητικότητα ή ευστάθεια
- είναι εφαρμόσιμη σε κάθε σύστημα ανεξαρτήτως της εσωτερικής του δομής και μορφολογίας

Καταρχήν, επισημαίνουμε πως η συγκεκριμένη μέθοδος λειτουργεί πολύ αποδοτικά για μικρής τάξης μεγέθους συστήματα (μερικές εκατοντάδες καταστάσεις) επιστρέφοντας ακριβέστερα αποτελέσματα συγκριτικά με τους υποχώρους Krylov. Για μεγαλύτερης κλίμακας συστήματα όμως, η μέθοδος αποδίδει καλύτερα όταν συνδυάζεται με κάποια από τις τεχνικές που περιγράψαμε παραπάνω. Με άλλα λόγια, πρόκειται για μια διαδικασία ελάττωσης της τάξης μοντέλου, η οποία αποτελείται από 2 βήματα. Κατά το πρώτο βήμα, χρησιμοποιούμε κάποια από τις παραπάνω τεχνικές για να μειώσουμε την υπερβολικά μεγάλη τάξη μεγέθους σε μερικές εκατοντάδες και εν συνεχεία δρούμε πάνω σε αυτό το ήδη μειούμενο μοντέλο εφαρμόζοντας την TBR, ως ένα βήμα βελτιστοποίησης της τάξης του τελικού μοντέλου. Οι παραπάνω τεχνικές που περιγράψαμε είναι υπολογιστικά φθηνότερες, αλλά όχι τόσο αποδοτικές και βέλτιστες όσο η TBR. Η TBR εφαρμόζεται σε όσο το δυνατόν μικρότερης τάξης συστήματα, καθώς διαφορετικά λόγω της υψηλής χρονικής πολυπλοκότητας των υπολογισμών της, η αναγωγή θα ήταν υπερβολικά χρονοβόρα έως και ανέφικτη [1].

Εξαιτίας της πολυπλοκότητας της υλοποίησης και με δεδομένο ότι πραγματοποιείται στα πλαίσια προπτυχιακής διπλωματικής εργασίας θα επικεντρώσουμε το ενδιαφέρον μας μόνο στο δεύτερο στάδιο αυτής της διττής διαδικασίας υπολογισμού μοντέλων μειωμένης τάξης, δηλαδή στην μέθοδο TBR.

Αναλυτικότερα, ένα ιδιαίτερο χαρακτηριστικό του συγκεκριμένου αλγορίθμου είναι ότι προσφέρει υπολογίσιμα όρια σφάλματος μεταξύ της συνάρτησης μεταφοράς του αρχικού και του τελικού, ελαττωμένου μοντέλου, γεγονός που αποτελεί ένδειξη της ορθότητας της απόκρισης. Συγκεκριμένα, η άπειρη νόρμα της διαφοράς της τελικής από την αρχική συνάρτηση μεταφοράς είναι άνω φραγμένη από το διπλάσιο του αθροίσματος των  $N-q-1$  μικρότερων ιδιοτιμών, όπου  $q$  η νέα τάξη του συστήματος [1]. Δηλαδή:

$$\|H(s) - H_q(s)\|_{\infty} \leq 2 \sum_{k=q+1}^N \sigma_k \quad (3.6)$$

Συνάμα, η βασική εκδοχή (Standard Approach) της TBR μεθόδου για συμμετρικά συστήματα καθώς και ορισμένες παραλλαγές της εγγυώνται την διατήρηση της παθητικότητας στο νέο, μειούμενο μοντέλο, κάτι για το οποίο δεν μας παρέχεται καμμία βεβαιότητα στις υπόλοιπες μεθόδους MOR. Οι παραλλαγές του αλγορίθμου TBR δεν θα μας απασχολήσουν στα πλαίσια αυτής της εργασίας, ωστόσο όμως θα αναφέρουμε τις επιπλέον εγγυήσεις που δίνουν σε σχέση με την βασική προσέγγιση και θα θέσουμε ως μελλοντικό στόχο την εξέτασή τους.

Ας υποθέσουμε τώρα ότι διαθέτουμε ένα καταστατικό, συμμετρικό μοντέλο, δηλαδή:

$$E \frac{dx}{dt} = Ax(t) + Bu(t) \quad (3.7)$$

$$y(t) = Cx(t) + Du(t) \quad (3.8)$$

για το οποίο ισχύει ότι  $E, A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times p}$ ,  $C \in \mathbb{R}^{p \times n}$ ,  $D \in \mathbb{R}^{p \times p}$  και ότι  $y(t), u(t) \in \mathbb{R}^p$ . Επιπλέον, λόγω της συμμετρίας ισχύουν:

- ✓  $A = A^T$
- ✓  $B = C^T$
- ✓  $A$  αρνητικά ημι - ορισμένος, δηλαδή  $A \leq 0$
- ✓  $E = I$
- ✓  $D \geq 0$

Εφόσον,  $\text{Re}\{j\omega I - A\} = \frac{-1}{2}(A + A^T) = -A \geq 0$ , αφού ο  $A$  αρνητικά ημι ορισμένος, το αρχικό μοντέλο συστήματος θα είναι positive - real. Με επισκόπηση των Lyapunov εξισώσεων και με τα παραπάνω δεδομένα, προκύπτει:

$$\begin{aligned} \begin{cases} AW_c + W_c A^T = -BB^T \\ A^T W_o + W_o A = -C^T C \end{cases} & \xLeftrightarrow{B=C^T} \begin{cases} AW_c + W_c A^T = -C^T C \\ A^T W_o + W_o A = -C^T C \end{cases} \\ \xLeftrightarrow{A=A^T} \begin{cases} AW_c + W_c A = -C^T C \\ AW_o + W_o A = -C^T C \end{cases} & \Leftrightarrow \begin{cases} 2AW_c = -C^T C \\ 2AW_o = -C^T C \end{cases} \\ & \xLeftrightarrow[\text{διαίρεση κατά μέλη}] W_c = W_o \end{aligned} \quad (3.9)$$

Επίσης, θα ισχύει  $T^{-1} = T^T$  και συνεπώς ο μετασχηματισμός ομοιότητας είναι ένας μαθηματικός μετασχηματισμός. Κατά συνέπεια, ο  $\tilde{A}$  πρέπει να είναι αρνητικά ημι - ορισμένος και  $\tilde{B} = \tilde{C}^T$ . Με άλλα λόγια, το νέο σύστημα θα είναι positive - real (PR), δηλαδή παθητικό μοντέλο και κατά συνέπεια θα διατηρείται η ιδιότητα που είχε το αρχικό μοντέλο. Παρατηρούμε έτσι πως η μοναδική απαίτηση που θέτει ο αλγόριθμος TBR για να διατηρήσει τις ιδιότητες του αρχικού μοντέλου και στο τελικό, είναι το αρχικό μοντέλο να είναι συμμετρικό.

Η μέθοδος TBR εφαρμόζεται σε κάθε είδους σύστημα ανεξάρτητα από την εσωτερική του δομή, εν αντιθέσει με τις μεθόδους Krylov που εφαρμόζονται μόνο σε κυκλώματα RLC.

Σε αυτό το σημείο κρίνεται απαραίτητο να προχωρήσουμε ένα βήμα παραπέρα και να περιγράψουμε την φυσική ερμηνεία του αλγορίθμου, ώστε να τον κατανοήσουμε και διαισθητικά [1]. Σε πιο αφαιρετικό επίπεδο, ο στόχος του αλγορίθμου είναι να μειώσει την τάξη του μαθηματικού μοντέλου του συστήματος, περικλύποντας ουσιαστικά τις καταστάσεις εκείνες που θεωρείται πολύ δύσκολο να κριθούν είτε ως ελέγξιμες είτε ως παρατηρήσιμες.

Με άλλα λόγια, διατηρούμε τις καταστάσεις, οι οποίες είναι τόσο ελέγξιμες, δηλαδή δεν απαιτείται να τους δοθεί μια υπερβολικά μεγάλη νόρμα εισόδου για να



ξεκινήσουν την λειτουργία τους, όσο και παρατηρήσιμες που σημαίνει ότι παράγουν μεγάλες εξόδους. Όλες οι υπόλοιπες καταστάσεις θεωρούμε ότι δεν συνεισφέρουν καθόλου στην απόκριση του συστήματος και συνεπώς μπορούμε να τις αφαιρέσουμε, ώστε να επιταχύνουμε τους υπολογισμούς της απόκρισης. Στο τελικό μοντέλο, συμπεριλαμβάνουμε μόνο τις καταστάσεις που είναι ταυτοχρόνως ελέγξιμες και παρατηρήσιμες<sup>4</sup> και αυτός ο αριθμός καθορίζει και την τάξη του καινούριου συστήματος.

Το διάνυσμα ελεγχιμότητας  $W_c$  αναφέρεται στην είσοδο εκείνη του συστήματος που δίνει την ελάχιστη νόρμα  $L_2$  και μπορεί να ελέγξει το σύστημα στην κατάσταση  $x_0$  την χρονική στιγμή 0.

$$x_0^T W_c^{-1} x_0 = \inf \left\{ \int_{-\infty}^0 u(t)^T u(t) dt, u(t) \text{ controlling to } x(0) = x_0 \right\} \quad (3.10)$$

Το διάνυσμα παρατηρησιμότητας  $W_o$  αναφέρεται στην  $L_2$  νόρμα της εξόδου που παράγεται όταν  $u(t) = 0$ ,  $\forall t$  από μια αρχική κατάσταση  $x_0$  την χρονική στιγμή 0.

$$x_0^T W_o x_0 = \int_0^{\infty} y(t)^T y(t) dt, x(0) = x_0, u(t) = 0, \forall t \geq 0 \quad (3.11)$$

Εποπτικά, με βάση τις τελευταίες 2 εξισώσεις, ισχύει:

- Εάν η κατάσταση  $X_0$  είναι δύσκολο να προσεγγιστεί από κάποια είσοδο, τότε η ποσότητα  $x_0^T W_c^{-1} x_0$  είναι πολύ μεγάλη και πρακτικά μη ελέγξιμη.
- Εάν η ποσότητα  $x_0^T W_o x_0$  είναι μικρή, τότε η επίδραση της κατάστασης  $X_0$  στην έξοδο είναι μικρή και κατ' επέκταση το σύστημα πολύ δύσκολα παρατηρήσιμο.

<sup>4</sup> Η συνάρτηση μεταφοράς θα βρίσκεται στην ελάχιστη μορφή της, αν και μόνο αν το σύστημα είναι ταυτοχρόνως ελέγξιμο και παρατηρήσιμο.

Η σελίδα αυτή είναι σκόπιμα λευκή.

# ΚΕΦΑΛΑΙΟ 4

## Ανάπτυξη Λογισμικού MATLAB

Το περιβάλλον που χρησιμοποιήθηκε για την υλοποίηση της προσομοίωσης του αλγορίθμου μείωσης τάξης μοντέλου, Truncated Balanced Realization, είναι το MATLAB<sup>5</sup> σε έκδοση 8 (R2012b). Στο παρόν κεφάλαιο περιγράφουμε συνοπτικά το γραφικό περιβάλλον αυτού του λογισμικού.

### 4.1 Περιβάλλον Ανάπτυξης Αλγορίθμου

Το MATLAB παρέχει ένα δυναμικό και εύχρηστο υπολογιστικό περιβάλλον για υλοποίηση επιστημονικών εφαρμογών σε ένα μεγάλο φάσμα πεδίων, όπως στη Γραμμική Άλγεβρα, Στατιστική, Αριθμητική Ανάλυση και Επιστημονικό Υπολογισμό, Επεξεργασία Σημάτων και Εικόνας, Θεωρία Ελέγχου και Γραφικά. Έχει υλοποιηθεί σε πολλές λειτουργικές πλατφόρμες (όπως Windows, Linux, MAC OS, UNIX) και δύο βασικές εκδόσεις, την επαγγελματική και την εκπαιδευτική (student edition). Σε πιο αφαιρετικό επίπεδο μπορούμε να θεωρήσουμε ότι αποτελεί ένα λογισμικό πολύ υψηλής απόδοσης κατάλληλο για σύνθετες υπολογιστικές διαδικασίες, οι οποίες με το χέρι θα ήταν τρομερά χρονοβόρες και ταυτόχρονα θα εγκυμονούσε ο κίνδυνος λαθών. Μερικές από τις εφαρμογές του αναφέρονται ακολούθως:

- ταχύτητα υπολογιστικών αναλύσεων
- δυνατότητα προσομοίωσης φυσικών συστημάτων
- δυνατότητα υλοποίησης αλγορίθμων
- υψηλής ποιότητας γραφικές παραστάσεις και *animations*

Ενδεικτικά και για λόγους πληρότητας αξίζει να αναφέρουμε ότι το όνομα MATLAB αποτελεί συντομογραφία των λέξεων Matrix Laboratory. Η βασική δομική

---

<sup>5</sup> Matrix Laboratory

μονάδα του MATLAB όπως διαφαίνεται και από το όνομα του είναι ο πίνακας (Matrix) και ο θεμελιώδης τύπος δεδομένου είναι το διάνυσμα. Όλα τα υπόλοιπα στοιχεία που θα συναντήσει κανείς στο λογισμικό προκύπτουν ως συνδυασμός αυτών των δύο.

Πιο συγκεκριμένα, το MATLAB παρέχει στον χρήστη του ένα διαδραστικό περιβάλλον με μερικές χιλιάδες ενσωματωμένες έτοιμες συναρτήσεις με την βοήθεια των οποίων εκτελούνται από απλοί μαθηματικοί υπολογισμοί μέχρι και προγράμματα με εντολές παρόμοιες με αυτές που υποστηρίζει μια γλώσσα υψηλού επιπέδου. Συγκεκριμένα εκτελεί απλές μαθηματικές πράξεις, αλλά εξίσου εύκολα χειρίζεται μιγαδικούς αριθμούς, δυνάμεις, πίνακες, διανύσματα και πολυώνυμα. Μπορεί επίσης να αποθηκεύει και να ανακαλεί δεδομένα καθώς και να σχεδιάζει γραφικά. Παράλληλα, το MATLAB είναι ικανό να εκτελέσει σύνθετες πράξεις (όπως η εύρεση ορίζουσας ή αντιστρόφου) σε πίνακες μεγάλης διάστασης ( $n > 1000$ , όπου  $n$  η διάσταση του πίνακα) μέσα σε κλάσματα του δευτερολέπτου. Οι ενσωματωμένες συναρτήσεις του λογισμικού του παρέχουν μεταξύ των άλλων τα απαραίτητα πακέτα για υπολογισμούς γραμμικής άλγεβρας, ανάλυσης δεδομένων (Data Mining), επεξεργασίας σημάτων καθώς και επίλυσης διαφορικών εξισώσεων. Συνάμα, όμως το MATLAB παρέχει την δυνατότητα για προγραμματιστικές εργασίες (script language), αφού εμπεριέχει εντολές διακλάδωσης (“if”, “if-then-else”) και επαναληπτικές εντολές (“while”, “for”, “do-while”).

Ταυτόχρονα, ο χρήστης έχει την δυνατότητα να αναπτύξει τις δικές του συναρτήσεις με ονόματα που ο ίδιος θα ορίσει και με κατάληξη \*.m (M-files). Από την στιγμή αυτή και μετά οι συναρτήσεις αυτές ανήκουν στο δυναμικό των ενσωματωμένων συναρτήσεων της MATLAB. Όλες οι μεταβλητές που δημιουργούνται μέσα στο αρχείο που περιέχει τη συνάρτηση είναι απομονωμένες (τοπικές μεταβλητές) από το workspace της MATLAB. Η μόνη σύνδεση μεταξύ των τοπικών μεταβλητών και του workspace είναι οι μεταβλητές εισόδου και εξόδου. Αν μία συνάρτηση αλλάζει την τιμή οποιασδήποτε από τις μεταβλητές εισόδου, οι αλλαγές δεν επηρεάζουν τις μεταβλητές του workspace.

Επιπλέον, παρέχονται από το λογισμικό πολλές εργαλειοθήκες (Toolboxes), οι οποίες προορίζονται για την δημιουργία πιο εξειδικευμένων εφαρμογών. Ενδεικτικά, αναφέρουμε μερικούς όπως συμβολικούς υπολογισμούς, επεξεργασία εικόνας (Image Processing), σχεδιασμός και αναπαράσταση συστημάτων ελέγχου (Control System Toolbox), νευρωνικά δίκτυα (Neural Networks), ασαφή λογική (Fuzzy Logic).

Το MATLAB μπορεί ακόμη να διαβάσει διάφορους τύπους αρχείων, όπως:

- M-files: Είναι ASCII αρχεία κειμένου με την κατάληξη.m. Διακρίνονται σε script files και σε function files.

- I. Script files: Αποτελούν ένα σύνολο από εντολές που εκτελούνται μέσα από το Command Window, δίνοντας μόνο το όνομα του αρχείου. Οι μεταβλητές τους έχουν καθολικό χαρακτήρα (global variables) και καταγράφονται μέσα στο Workspace πάντα. Είναι χρήσιμα στις περιπτώσεις όπου ο

χρήστης πρέπει να επαναλάβει μια συγκεκριμένη ακολουθία εντολών και υπολογισμών με διαφορετικές εισόδους κάθε φορά.

- II. Function files: Οι μεταβλητές των αρχείων συνάρτησης είναι τοπικές (local). Μοιάζουν με τις συναρτήσεις που χρησιμοποιούμε στις γλώσσες προγραμματισμού C, Java. Το όνομα της συνάρτησης πρέπει να είναι ίδιο με το όνομα του αρχείου m-file, χωρίς την κατάληξη .m φυσικά. Π.χ. έγκυρος ορισμός μιας συνάρτησης είναι:

```
function [rsys, hsv] = tbr (sys,order);
```

- Mat-files: Είναι binary data files με την κατάληξη .mat. Δημιουργούνται όταν αποθηκεύουμε δεδομένα με την εντολή save.
- Fig-files: Είναι binary figure αρχεία και δημιουργούνται όταν ο χρήστης αποθηκεύει ένα γράφημα.
- Mex-files: Κατασκευάζονται κατά την σύνδεση του MATLAB με άλλες γλώσσες προγραμματισμού όπως η C, Fortran.

Το MATLAB παρέχει επίσης την δυνατότητα εισαγωγής δεδομένων (text ή binary data) από ένα αρχείο (Πίνακας 4.1) και εξαγωγής δεδομένων (Πίνακας 4.2) σε αρχείο (text ή binary data). Υπάρχουν ποικίλοι τρόποι που μπορεί να υλοποιηθεί κάτι τέτοιο και εξαρτάται από το είδος της εφαρμογής και το είδος αρχείου που θέλουμε να εισάγουμε ή να εξάγουμε. Οι παρεχόμενες από το λογισμικό συναρτήσεις για εισαγωγή text data είναι:

Συνάρτηση	Τύπος Δεδομένων
load	Αριθμητικά δεδομένα
fscanf	Αριθμητικά και αλφαριθμητικά δεδομένα
textread	Αριθμητικά και αλφαριθμητικά δεδομένα

**Πίνακας 4.1:** Συναρτήσεις για εισαγωγή δεδομένων

Ενδεικτικές συναρτήσεις για εξαγωγή text data αποτελούν:

Συνάρτηση	Τύπος Δεδομένων
fprintf	Αριθμητικά και αλφαριθμητικά δεδομένα
save	Αριθμητικά δεδομένα
diary	Αριθμητικά δεδομένα και διατάξεις κελιών

**Πίνακας 4.2:** Συναρτήσεις για εξαγωγή δεδομένων

## 4.2 Το περιβάλλον της MATLAB

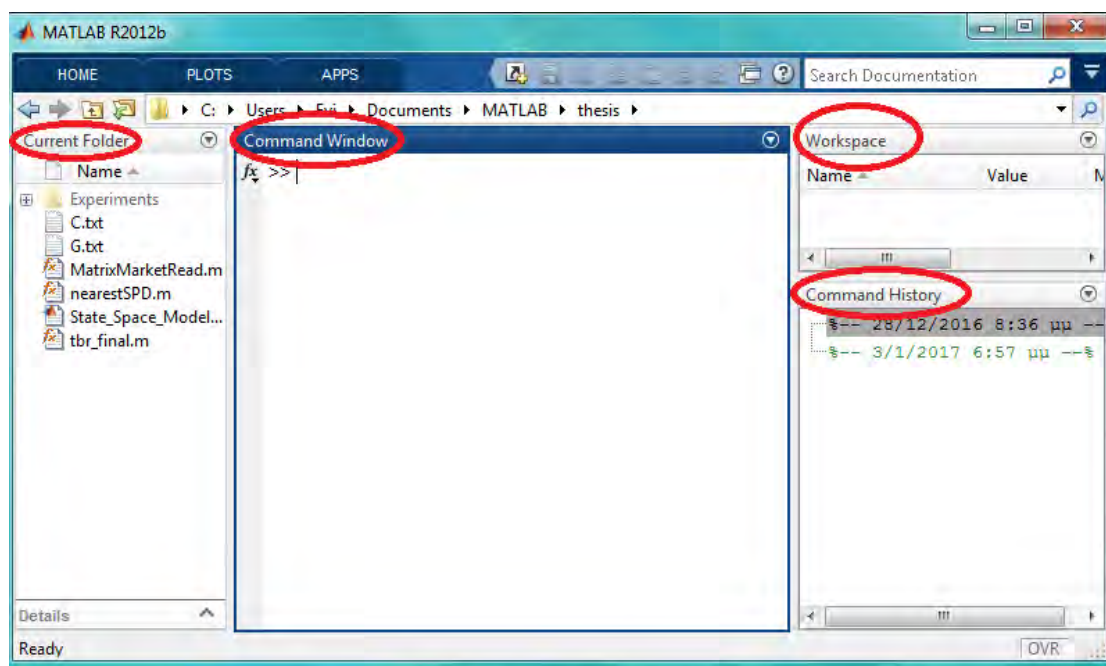
Το λογισμικό μπορεί πολύ εύκολα όπως περιγράψαμε και παραπάνω να εγκατασταθεί και να χρησιμοποιηθεί στα περισσότερα λειτουργικά συστήματα που κυκλοφορούν στην αγορά. Ανεξαρτήτως λοιπόν του λειτουργικού συστήματος που θα επιλέξει κανένας για να εγκαταστήσει το λογισμικό θα συναντήσει τα εξής 3 διαφορετικά είδη παραθύρων τα οποία αναλύονται ακολούθως.

### 1. Επιφάνεια Εργασίας του MATLAB (Desktop MATLAB)

Με το που εκκινήσει το MATLAB εμφανίζονται 4 υπό - παράθυρα: Παράθυρο Εντολών (Command Window), Παράθυρο Ιστορικού (Command History), Παράθυρο χώρου εργασίας (Workspace), Παράθυρο Τρέχοντος Καταλόγου (Current Folder Window) και τα οποία περιγράφονται με λίγη περισσότερη λεπτομέρεια ευθύς αμέσως (Εικόνα 18).

- Παράθυρο Εντολών (Command Window): Αποτελεί το βασικό παράθυρο και τον χώρο τον οποίο χρησιμοποιεί ο χρήστης για να εισάγει τα απαραίτητα δεδομένα, να εκτελεστούν οι υπολογισμοί και να εξαχθούν τα επιζητούμενα αποτελέσματα. Ουσιαστικά αποτελεί τον χώρο αλληλεπίδρασης του χρήστη με το λογισμικό. Χαρακτηρίζεται από το σύμβολο προτροπής (>>, MATLAB command prompt).
- Παράθυρο Ιστορικού (Command History): Το σύνολο των εντολών, οι οποίες πληκτρολογούνται στο Παράθυρο εντολών, μετά από κάθε ένα σύμβολο προτροπής, καταγράφονται στο παρών Παράθυρο ιστορικού εντολών και μπορούν ανά πάσα χρονική στιγμή να επανεκτελεστούν με απλό Drag and Drop στο Παράθυρο Εντολών. Στο παράθυρο αυτό υπάρχουν καταχωρημένες εντολές, οι οποίες έχουν εκτελεστεί στο πλαίσιο πολυσύνθετων τμημάτων, ακόμα και μέρες πριν από την τελευταία εισαγωγή στο σύστημα.
- Παράθυρο χώρου εργασίας (Workspace): Στο παράθυρο αυτό, απεικονίζονται όλες οι μεταβλητές, οι οποίες εισάγονται και χρησιμοποιούνται στο Παράθυρο εντολών και οι οποίες είναι αποθηκευμένες στην μνήμη του υπολογιστή. Στο παρών παράθυρο παρέχονται πληροφορίες για τον τύπο και το μέγεθος κάθε μεταβλητής. Ο χρήστης έχει τη δυνατότητα να ανακαλέσει ανά πάσα στιγμή τη μεταβλητή αυτή στο Παράθυρο εντολών, πληκτρολογώντας απλώς το όνομα της αλλά και να σχεδιάσει γραφική παράσταση της μεταβλητής.
- Παράθυρο Τρέχοντος Καταλόγου (Current Folder Window): Είναι το σημείο της επιφάνειας εργασίας του MATLAB, όπου αναγράφονται το σύνολο των αρχείων, τα οποία είναι αποθηκευμένα στον τρέχον κατάλογο (Directory) του συστήματος. Εάν ένα m-File βρίσκεται μέσα στον φάκελο τότε και μόνον τότε

μπορεί να κληθεί. Παρέχεται η δυνατότητα πλοήγησης μέσα σε αυτό, όπως επίσης με τη χρήση του ποντικιού (κάνοντας δεξί κλικ με το ποντίκι πάνω στο επιλεγμένο αρχείο), είναι δυνατή η εκτέλεση διάφορων επιλογών, οι οποίες σχετίζονται με το αρχείο (μετονομασία αρχείου, διαγραφή αρχείου, εκτέλεση M- File).



Εικόνα 18: Περιβάλλον επιφάνειας εργασίας του MATLAB

## 2. Παράθυρο σύνταξης (Editor Window)

Είναι το παράθυρο, στο οποίο ο χρήστης μπορεί να αναπτύξει, να επεξεργαστεί, να αποθηκεύσει τα δικά του αρχεία εντολών, τα οποία κυρίως απαρτίζονται από τα M-Files (scripts, functions). Αν και είναι δυνατό τα αρχεία αυτά να συνταχθούν μέσω των κλασσικών προγραμμάτων σύνταξης (text editors), το MATLAB προτιμάται, αφού προσφέρει το αντίστοιχο πρόγραμμα, το οποίο είναι ενσωματωμένο στο πακέτο λογισμικού.

## 3. Παράθυρο γραφημάτων (Figure Window)

Στο συγκεκριμένο και ξεχωριστό παράθυρο από τα υπόλοιπα συμπεριλαμβάνονται όλα τα γραφήματα που έχουν προκύψει από αντίστοιχες εντολές στο παράθυρο εντολών. Μάλιστα, δίνεται η δυνατότητα στο pop-up παράθυρο με την γραφική παράσταση που ανοίγει να επεξεργαστούμε τις ιδιότητες του επικείμενου γραφήματος.

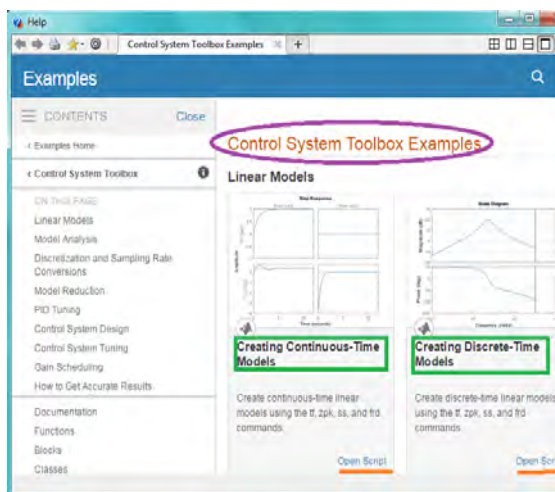
### 4.3 Control System Toolbox

Το βασικό πρόγραμμα της MATLAB σε συνδυασμό με το εξειδικευμένο εργαλείο Control System Toolbox δίνει την δυνατότητα χρήσης του λογισμικού στην σχεδίαση και την ανάλυση των συστημάτων αυτομάτου ελέγχου. Επίσης, παρέχει εργαλεία κατάλληλα για προσομοίωση συστημάτων. Το πακέτο προσφέρει δομές δεδομένων για την περιγραφή κοινών μορφών αναπαράστασης δυναμικών συστημάτων, όπως η καταστατική μορφή ή η συνάρτηση μεταφοράς. Εντολές όπως 'ss', 'tf' έχουν χρησιμοποιηθεί για αυτόν τον σκοπό.

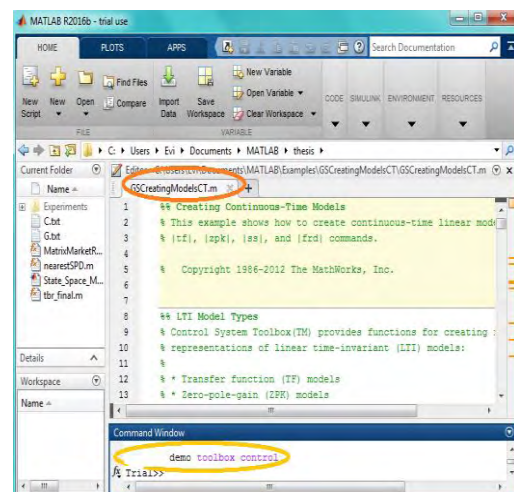
Χρησιμοποιήθηκε η έκδοση 10.1 (R2016b). Με την βοήθεια της εντολής 'help control' στο παράθυρο εντολών μπορούμε να δούμε διάφορες κατηγορίες εντολών για μοντελοποίηση, ανάλυση, σχεδιασμό δυναμικών συστημάτων που υποστηρίζει το εργαλείο καθώς και τον τρόπο σύνταξής τους. Το Control System Toolbox περιέχει έτοιμες συναρτήσεις κατάλληλες για:

- Μοντελοποίηση, δηλαδή δημιουργία μαθηματικών μοντέλων για ΓΧΑ συστήματα (c2d, ss2ss, minreal, hsvd, modred).
- Ανάλυση συστημάτων τόσο στο πεδίο του χρόνου όσο και στο πεδίο της συχνότητας (isstable, ctrb, obsv, gram).
- Σχεδιασμό συστημάτων ελέγχου (zero, plot).
- Δημιουργία γραφικών παραστάσεων για την απόκριση συχνότητας (step, impulse, bode, bodemag, norm).

Επιπροσθέτως, πληκτρολογώντας στο παράθυρο εντολών "demo toolbox control" παίρνουμε μια λίστα με διαθέσιμα παραδείγματα και επιλέγοντας 'Open Script' μπορούμε να δούμε τα αντίστοιχα κομμάτια κώδικα (Εικόνες 19, 20 αντίστοιχα).



Εικόνα 19: Demo χρήσης Toolbox



Εικόνα 20: Script μοντελοποίησης



Η σελίδα αυτή είναι σκόπιμα λευκή.

# ΚΕΦΑΛΑΙΟ 5

## Υλοποίηση και βήματα αλγορίθμου TBR

Οι μέθοδοι υποβιβασμού τάξης μοντέλου (MOR) είναι τεχνικές οι οποίες αποσκοπούν στη μείωση της υπολογιστικής πολυπλοκότητας ή της τάξης των μαθηματικών μοντέλων με απώτερο στόχο την ταχύτερη προσομοίωση τους. Στο παρόν κεφάλαιο θα περιγράψουμε και θα αναλύσουμε τα οχτώ βήματα του αλγορίθμου TBR.

### 5.1 Η μέθοδος υποβιβασμού τάξης μοντέλου TBR

Ας υποθέσουμε ότι έχουμε ένα μοντέλο συστήματος στην ακόλουθη καταστατική μορφή [1]:

$$E \frac{dx}{dt} = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

με  $E, A \in \mathbb{R}^{n \times n}$  και  $E=I$ ,  $B \in \mathbb{R}^{n \times p}$ ,  $C \in \mathbb{R}^{p \times n}$ ,  $D \in \mathbb{R}^{p \times p}$ , όπου με  $n$  συμβολίσαμε την διάσταση του αρχικού συστήματος. Επίσης, θεωρούμε ότι  $y(t), u(t) \in \mathbb{R}^p$ . Ακόμη,  $\det(E) \neq 0$ , δηλαδή ο  $E$  είναι αντιστρέψιμος πίνακας. Ο στόχος των αλγορίθμων μείωσης τάξης μοντέλου είναι να παράγουν ένα νέο μοντέλο που θα βρίσκεται επίσης σε καταστατική μορφή:

$$\tilde{E} \frac{d\tilde{x}}{dt} = \tilde{A}\tilde{x}(t) + \tilde{B}u(t)$$

$$\tilde{y}(t) = \tilde{C}\tilde{x}(t) + \tilde{D}u(t)$$

με  $\tilde{E}, \tilde{A} \in \mathbb{R}^{q \times q}$ ,  $\tilde{B} \in \mathbb{R}^{q \times p}$ ,  $\tilde{C} \in \mathbb{R}^{p \times q}$ ,  $\tilde{D} \in \mathbb{R}^{p \times p}$ , όπου  $q$  η διάσταση του καινούριου μειωμένου μοντέλου με  $q \ll n$ . Για τις συναρτήσεις μεταφοράς τους θα ισχύει αντίστοιχα ότι:

$$H(s) = C(sE - A)^{-1}B + D$$

$$\tilde{H}(s) = \tilde{C} (s\tilde{E} - \tilde{A})^{-1}\tilde{B} + \tilde{D}$$

Η άπειρη νόρμα της διαφοράς των δύο συναρτήσεων μεταφοράς θα είναι άνω φραγμένη από κάποιο όριο  $\varepsilon$ :

$$\|H(s) - \tilde{H}(s)\|_{\infty} < \varepsilon$$

όπου  $s$  μεταβλητή στο πεδίο της συχνότητας και  $\varepsilon$  επιτρεπτό όριο σφάλματος.

Ως πρωταρχικό στάδιο, πρέπει να δημιουργήσουμε το σύστημα που αποτελείται από τους πίνακες  $A, B, C, D$ , το οποίο θα εισάγουμε ως όρισμα στον αλγόριθμο TBR. Ο πίνακας  $A$  θα είναι:

$$A = -R^{-1}G$$

όπου  $R, G$  έχουν διαβαστεί από αρχείο περιγραφής κυκλώματος και μετατραπεί σε πίνακες από την συνάρτηση `MatrixMarketRead.m`. Είναι επιτακτικό για τον πίνακα  $R$  να ορίζεται ο αντίστροφος του για να μπορέσουμε να συνεχίσουμε. Επιπλέον, στο σημείο αυτό πρέπει να διαπιστώσουμε αν ο πίνακας  $A$  είναι ευσταθής, ελέγχοντας αν οι ιδιοτιμές του βρίσκονται στο αριστερό μιγαδικό ημιεπίπεδο. Εν συνεχεία, ο πίνακας  $BB$  αντιστοιχίζει το διάνυσμα των εισόδων στο διάνυσμα κατάστασης, τοποθετώντας μονάδα στις θέσεις του πίνακα που θεωρούμε θύρες, έτσι ώστε  $BB = C^T$ . Θα είναι διάστασης  $n \times N_i$ , όπου  $n$  οι μεταβλητές του διανύσματος κατάστασης και  $N_i$  ο αριθμός των εισόδων. Ο πίνακας  $B$  που θα εισάγουμε ως όρισμα στον TBR, μετά από την αναγωγή του συστήματος σε καταστατική μορφή, θα είναι:

$$B = R^{-1}BB$$

Ο πίνακας  $C$  αντιστοιχίζει το διάνυσμα της εξόδου στο διάνυσμα κατάστασης και είναι διάστασης  $N_o \times n$ , όπου  $n$  οι μεταβλητές κατάστασης και  $N_o$  ο αριθμός των εξόδων. Είναι ίσος με τον ανάστροφο πίνακα του  $BB$ , δηλαδή:

$$C = (BB)^T$$

Ο πίνακας  $D$  είναι διάστασης  $N_i \times N_o$  και τις περισσότερες φορές όλες του οι εγγραφές είναι μηδέν.

Ο αλγόριθμος TBR που θα υλοποιήσουμε αποτελείται από 2 μέρη. Το πρώτο μέρος περιλαμβάνει την εύρεση του μετασχηματισμού ομοιότητας που θα επιδράσει στο αρχικό σύστημα και θα αλλάξει την τάξη του. Συνοπτικά, ο μετασχηματισμός προκύπτει από την επίλυση των 2 εξισώσεων Lyapunov, την παραγοντοποίησή τους κατά Cholesky και την αποσύνθεση σε Singular Values του τελευταίου γινομένου. Το δεύτερο μέρος εστιάζει στην επιλογή της καινούριας, μικρότερης τάξης μοντέλου  $q$  με βάση τις Hankel Singular Values και στον διαμερισμό των τροποποιημένων πινάκων

σε ισχυρούς και αδύναμους υποπίνακες ιδιόμορφων τιμών. Στο τέλος, κρατούμε τις  $q$  ενεργειακά αποδοτικότερες καταστάσεις, περικόπτοντας τις μικρές, ασήμαντες ιδιόμορφες τιμές και τερματίζει ο αλγόριθμος. Ακολούθως, περιγράφονται αναλυτικά τα δύο μέρη.

## 5.2 Υλοποίηση TBR

### 5.2.1 Balanced Realization

Συνοπτικά, τα βήματα του αλγορίθμου σε μορφή ψευδοκώδικα έχουν ως εξής [1],[5]:

---

#### Αλγόριθμος 1: Truncated Balanced Realization (TBR)

---

- 1: Λύσε την εξίσωση  $AW_c + W_cA^T + BB^T = 0$  ως προς  $W_c$ .
  - 2: Λύσε την εξίσωση  $A^T W_o + W_oA + C^TC = 0$  ως προς  $W_o$ .
  - 3: Εφάρμοσε παραγοντοποίηση Cholesky  $W_c = L_c L_c^T$ ,  $W_o = L_o L_o^T$ .
  - 4: Εφάρμοσε SVD στο γινόμενο Cholesky, δηλαδή  $L_o^T L_c = U \Sigma V$ , όπου  $\Sigma$  είναι θετικός, διαγώνιος πίνακας και  $U, V$  έχουν ορθοκανονικές στήλες.
  - 5: Υπολόγισε τους μετασχηματισμούς ομοιότητας,  $T = L_c V \Sigma^{-\frac{1}{2}}$ ,  $T^{-1} = \Sigma^{-\frac{1}{2}} U^T L_o^T$ .
  - 6: Υπολόγισε το ισορροπημένο μοντέλο συστήματος ως  $\hat{A} = T^{-1}AT$ ,  $\hat{B} = T^{-1}B$ ,  $\hat{C} = CT$ ,  $\hat{D} = D$ .
  - 7: Επέλεξε την νέα τάξη συστήματος και διαμέρισε τους πίνακες  $\hat{A}$ ,  $\hat{B}$ ,  $\hat{C}$ ,  $\hat{D}$  κατάλληλα.
  - 8: Περιέκοψε τους πίνακες  $\hat{A}$ ,  $\hat{B}$ ,  $\hat{C}$ ,  $\hat{D}$  για να σχηματιστεί το νέο μοντέλο συστήματος  $\tilde{A}$ ,  $\tilde{B}$ ,  $\tilde{C}$ ,  $\tilde{D}$ .
- 

Κατόπιν, περιγράφονται εκτενέστερα.

Η μέθοδος μείωσης τάξης μοντέλου συστήματος που θα εξετάσουμε είναι η TBR και εφαρμόζεται ως επί το πλείστον σε γραμμικά, χρονικά αμετάβλητα συστήματα. Όπως ήδη έχουμε επισημάνει ο στόχος είναι να ελαττώσουμε τον αριθμό των καταστάσεων ελαχιστοποιώντας την επίδραση στην συνάρτηση μεταφοράς. Η βασική ιδέα του αλγορίθμου έγκειται σε 2 καίριες παρατηρήσεις που θα αναλύσουμε εφεξής.

Η πρώτη εκ των δύο βασίζεται στην εύρεση ενός μετασχηματισμού ομοιότητας, ο οποίος μετασχηματίζει το αρχικό σύστημα σε μία νέα ισορροπημένη μορφή (Balanced Realization form), όπου τα καινούρια διανύσματα ελεγχσιμότητας και παρατηρησιμότητας θα πρέπει να είναι συγχρόνως ίσα και διαγώνια. Με άλλα λόγια, επιθυμούμε να βρούμε μια άλλη αναπαράσταση του συστήματος, μέσω της οποίας θα επιτυγχάνουμε ίσα ποσά παρεχόμενης ενέργειας από την είσοδο στο διάνυσμα κατάστασης και από το διάνυσμα κατάστασης στην έξοδο. Μετρική αυτών των ποσοτήτων ενέργειας είναι τα διανύσματα ελεγχσιμότητας και παρατηρησιμότητας.

Ο μετασχηματισμός αυτός αφήνει αναλλοίωτες τις ιδιότητες του αρχικού συστήματος και είναι εφικτός, εφόσον οι αρχικοί πίνακες είναι συμμετρικοί και επομένως μπορούν να διαγωνοποιηθούν ταυτόχρονα. Τα διάνυσμα ελεγχιμότητας και παρατηρησιμότητας, τα οποία ονομάζουμε συμβολικά ως  $W_c$ ,  $W_o$ , προκύπτουν από την λύση των εξής δύο εξισώσεων Lyapunov<sup>6</sup>:

$$AW_c + W_c A^T + BB^T = 0 \quad (5.1)$$

$$A^T W_o + W_o A + C^T C = 0 \quad (5.2)$$

όπου ορίζουμε ως  $A = -R^{-1}G$  και ως  $B$  και  $C$  τους πίνακες που αντιστοιχίζουν την είσοδο στο διάνυσμα κατάστασης και την έξοδο στο διάνυσμα κατάστασης, αντίστοιχα. Ορίστηκαν κατάλληλα στο εδάφιο 5.1.

Βασική προϋπόθεση για να λειτουργήσει ο αλγόριθμος αποτελεί, το σύστημα που εισάγεται ως όρισμα, να είναι ευσταθές, δηλαδή το πραγματικό μέρος των ιδιοτιμών του να βρίσκεται στο αρνητικό μιγαδικό ημιεπίπεδο, να είναι ελέγξιμο και παρατηρήσιμο καθώς και να βρίσκεται στην ελάχιστη μορφή του, δηλαδή να έχουμε απαλείψει κοινούς πόλους και μηδενικά από τον παρονομαστή και τον αριθμητή αντίστοιχα. Κάτω από αυτές τις προδιαγραφές οι παραπάνω δύο εξισώσεις Lyapunov θα έχουν μία και μοναδική λύση, τους θετικά ορισμένους πίνακες  $W_c$ ,  $W_o$ .

Ακολουθώντας, αποδεικνύουμε με μαθηματικό τρόπο ότι οι πίνακες αυτοί πράγματι επαληθεύουν τις εξισώσεις Lyapunov. Αρχικά, ας θεωρήσουμε την Lyapunov εξίσωση (5.1):

$$AW_c + W_c A^T + BB^T = 0$$

Θα αποδείξουμε ότι το διάνυσμα ελεγχιμότητας  $W_c = \int_0^\infty e^{At} BB^T e^{A^T t} dt$  επαληθεύει την εξίσωση (5.1).

$$\text{Ισχύει,} \quad A \int_0^\infty e^{At} BB^T e^{A^T t} dt + \int_0^\infty e^{At} BB^T e^{A^T t} dt A^T + BB^T = 0 \quad \Leftrightarrow$$

$$\int_0^\infty (A e^{At} BB^T e^{A^T t} dt + e^{At} BB^T e^{A^T t} A^T dt) + BB^T = 0 \quad \Leftrightarrow$$

$$\int_0^\infty \frac{d}{dt} (e^{At} BB^T e^{A^T t}) + BB^T = 0 \quad \Leftrightarrow$$

$$\lim_{t \rightarrow \infty} (e^{At} BB^T e^{A^T t}) - \lim_{t \rightarrow 0} (e^{At} BB^T e^{A^T t}) + BB^T = 0 \quad \Leftrightarrow$$

<sup>6</sup> **Aleksandr Mikhailovich Lyapunov** γεννήθηκε το 1857 στην Ρωσία. Πετυχημένος μαθηματικός, μηχανικός και φυσικός.

Ο πίνακας  $A$  είναι ευσταθής και συνεπώς όλες οι ιδιοτιμές του έχουν αρνητικό πραγματικό μέρος. Έτσι,  $\lim_{t \rightarrow \infty} (e^{At} B B^T e^{A^T t}) = 0$ .

Τελικά,

$$0 - e^{A0} B B^T e^{A^T 0} + B B^T = 0 \quad \Leftrightarrow$$

$$-B B^T + B B^T = 0$$

Με την ίδια ακριβώς διαδικασία μπορούμε να αποδείξουμε και ότι το διάνυσμα παρατηρησιμότητας  $W_o = \int_0^\infty e^{A^T t} C^T C e^{A t} dt$  επαληθεύει την δυϊκή εξίσωση Lyapunov:  $A^T W_o + W_o A + C^T C = 0$ .

Συνοπτικά, μπορούμε να παρατηρήσουμε ότι οι έννοιες των διανυσμάτων ελεγχιμότητας και παρατηρησιμότητας (Controllability, Observability Grammians) στην ουσία περιγράφουν την αλληλεπίδραση που υπάρχει σε ένα σύστημα μεταξύ του εξωτερικού περιβάλλοντος, δηλαδή των εισόδων και εξόδων και της εσωτερικής αναπαράστασης του, δηλαδή των μεταβλητών κατάστασης.

Η δεύτερη παρατήρηση σημαίνουσας αξίας έχει να κάνει με τις Hankel Singular Values (HSV), οι οποίες αποτελούν πιο καθοριστικό παράγοντα στην διαδικασία αναγωγής του αρχικού στο μειωμένο σύστημα. Προκύπτουν ως οι τετραγωνικές ρίζες των ιδιοτιμών του γινομένου των διανυσμάτων ελεγχιμότητας και παρατηρησιμότητας,  $W_c W_o$ .

$$\sigma_i = \sqrt{\lambda_i(W_c W_o)}, \quad i = 1, 2, \dots, n \quad \text{με } \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \quad (5.3)$$

Το κυριότερο χαρακτηριστικό τους εντοπίζεται στο ότι είναι ανεξάρτητες από την μορφή του συστήματος που μελετάμε, δηλαδή παραμένουν αμετάβλητες στο αρχικό και στο νέο σύστημα, χωρίς να επηρεάζονται από τον μετασχηματισμό που άλλαξε μεν την μορφή του συστήματος, αλλά όχι την συμπεριφορά του. Η συμπεριφορά του συστήματος είναι μετρική που καθορίζεται εξ' ολοκλήρου από τις εισόδους και τις αντίστοιχες εξόδους.

Ο σκοπός του αλγορίθμου είναι να διαμερίσουμε το σύστημα σε μπλοκ αποτελούμενα από Hankel Singular Values, διατηρώντας εκείνα που αντιπροσωπεύουν τις αριθμητικά μεγάλες ιδιόμορφες τιμές και αφαιρώντας όσα αντιστοιχούν σε μικρές και συνεπώς ασήμαντες για την απόκριση του συστήματος [6]. Οι Hankel Singular Values χρησιμοποιούνται για να υπολογίσουμε την νέα τάξη συστήματος.

Ως απόρροια των δύο αυτών παρατηρήσεων οδηγούμαστε στην εξής μαθηματικά ορισμένη μορφή:

$$\widetilde{W}_c = \widetilde{W}_o = \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} = \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_n \end{bmatrix}, \quad (5.4)$$

όπου ο όρος  $\Sigma_1$  αντιπροσωπεύει το “ισχυρό” υποσύστημα που διατηρείται και επιστρέφεται από τον αλγόριθμο, ενώ το  $\Sigma_2$  το “αδύναμο” υποσύστημα που δεν συνεισφέρει σε τίποτα στην απόκριση και άρα δεν αλλάζει κάτι εάν το αφαιρέσουμε.

Ως επόμενο βήμα του αλγορίθμου, εφαρμόζουμε ανάλυση Cholesky, με δεδομένο ότι οι πίνακες  $W_c$  και  $W_o$  είναι θετικά ορισμένοι (SPD – Semi Positive Defined) και συμμετρικοί. Αυτό σημαίνει ότι το σύστημα (A, B, C, D) πρέπει να είναι ελέγξιμο και παρατηρήσιμο, ώστε να προκύψουν θετικά ορισμένοι πίνακες. Επομένως, με την παραγοντοποίηση Cholesky, προκύπτει το γινόμενο ενός κάτω τριγωνικού και του ανάστροφου του πίνακα ως ακολούθως:

$$W_c = L_c L_c^T \quad (5.5)$$

$$W_o = L_o L_o^T \quad (5.6)$$

Κατόπιν, υπολογίζουμε τις ιδιόμορφες τιμές (Singular Value Decomposition, SVD) του γινομένου των άνω και κάτω τριγωνικών πινάκων  $L_c$  και  $L_o$ . Δηλαδή:

$$\begin{aligned} L_o^T L_c &= U \Sigma V = [U_1 \quad U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \\ &= [U_1 \quad U_2] \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}, \end{aligned} \quad (5.7)$$

όπου ο πίνακας  $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$  είναι πραγματικός, θετικά ορισμένος και διαγώνιος. Εμπεριέχει τις HSV τιμές σε αύξουσα διάταξη ( $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$ ), ενώ για τους ορθοκανονικούς πίνακες  $U, V$  ισχύει  $U^T U = I$  και  $V^T V = I$  αντίστοιχα.

Στην συνέχεια, με βάση τις τιμές που βρέθηκαν στα δύο προηγούμενα βήματα, υπολογίζουμε τον ευθύ και τον αντίστροφο μετασχηματισμό ομοιότητας  $T \in \mathbb{R}^{n \times n}$ ,  $T^{-1} \in \mathbb{R}^{n \times n}$  ως ακολούθως:

$$T = L_c V \Sigma^{-\frac{1}{2}} \quad (5.8)$$

$$T^{-1} = \Sigma^{-\frac{1}{2}} U^T L_o^T \quad (5.9)$$

όπου  $\Sigma^{-\frac{1}{2}} = \text{diag}(\frac{1}{\sqrt{\sigma_1}}, \dots, \frac{1}{\sqrt{\sigma_n}})$  περιέχει τις  $n$  HSV τιμές του συστήματος.

### 5.2.2 Διαδικασία Μείωσης Τάξης Μοντέλου

Οι μετασχηματισμοί ομοιότητας οι οποίοι χρησιμοποιούνται για να λάβουμε το νέο σύστημα, δίνουν τελικά τους καινούριους πίνακες:

$$\hat{A} = T^{-1}AT \quad (5.10)$$

$$\hat{B} = T^{-1}B \quad (5.11)$$

$$\hat{C} = CT \quad (5.12)$$

$$\hat{D} = D \quad (5.13)$$

Παράλληλα, τα διανύσματα ελεγχιμότητας και παρατηρησιμότητας μετασχηματίζονται ως ακολούθως:

$$\widehat{W_c} = T^{-1} W_c T^{-T} \quad (5.14)$$

$$\widehat{W_o} = T^T W_o T \quad (5.15)$$

Η επιλογή της τάξης  $q$  του νέου μοντέλου δεν είναι αυθαίρετη. Όπως έχουμε αναφέρει, έγκειται στις Hankel Singular Values. Πιο συγκεκριμένα, με δεδομένο κάποιο όριο σφάλματος που είτε το δίνει ο χρήστης δυναμικά από το πληκτρολόγιο είτε είναι δηλωμένο στατικά ( $\text{redErr} > 0$ ), ξεκινώντας από την μικρότερη ιδιόμορφη τιμή που βρίσκεται κάτω δεξιά στον διαγώνιο πίνακα  $\Sigma$  και κινούμενοι προς τις μεγαλύτερες, αθροίζουμε τις τιμές. Κάθε φορά προσθέτουμε την τιμή της τρέχουσας θέσης του πίνακα, στο άθροισμα όσων έχουν προσπελαστεί μέχρι στιγμής. Σε κάθε βήμα ελέγχουμε αν το τρέχον διπλάσιο άθροισμα των ιδιόμορφων τιμών είναι μικρότερο από το  $\text{redErr}$  που έχουμε ορίσει. Μόλις, το διπλάσιο άθροισμα των Hankel Singular Values που έχουν εξεταστεί, βρεθεί μεγαλύτερο από το  $\text{redErr}$ , σημαίνει ότι δεν μπορούμε να αφαιρέσουμε άλλες καταστάσεις ως ασήμαντες και η διαδικασία εύρεσης μειωμένης κλίμακας μοντέλου πρέπει να τερματιστεί. Συνεπώς, η τάξη  $q$  τίθεται ίση με τον τρέχον μετρητή προσαυξημένο κατά ένα, μιας και η επαναληπτική διαδικασία ξεκίνησε από την τάξη  $n$  του αρχικού συστήματος και κάθε φορά μειωνόταν κατά μία μονάδα.

Ας θεωρήσουμε ότι το σύστημα  $(A, B, C, D)$  διαμερίζεται ως ακολούθως:

$$\hat{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad \hat{C} = [C_1 \quad C_2]$$

- ✓ Ο υποπίνακας  $A_{11}$  είναι διάστασης  $q \times q$  και περιλαμβάνει τις εγγραφές από την πρώτη γραμμή έως και την γραμμή  $q$  και από την πρώτη στήλη έως και την στήλη  $q$ , όπου  $q$  η καινούρια διάσταση του συστήματος.



- ✓ Ο υποπίνακας  $A_{12}$  είναι διάστασης  $q \times (n-q)$  και περιλαμβάνει τις εγγραφές από την αρχή έως και την γραμμή  $q$  και από την στήλη  $q+1$  έως το τέλος.
- ✓ Ο υποπίνακας  $A_{21}$  είναι διάστασης  $(n-q) \times q$  και περιλαμβάνει τις εγγραφές από την γραμμή  $q+1$  έως το τέλος και από την πρώτη στήλη έως και την στήλη  $q$ .
- ✓ Ο υποπίνακας  $A_{22}$  είναι διάστασης  $(n-q) \times (n-q)$  και περιλαμβάνει τις εγγραφές από την γραμμή  $q+1$  έως το τέλος και από την στήλη  $q+1$  έως το τέλος.
- ✓ Αντίστοιχα, ο υποπίνακας  $B_1$  εμπεριέχει τις εγγραφές από την γραμμή 1 ως την γραμμή  $q$  και όλες τις στήλες, ενώ ο  $B_2$  τις εγγραφές από την γραμμή  $q+1$  ως το τέλος και όλες τις στήλες.
- ✓ Τέλος, ο υποπίνακας  $C_1$  εμπεριέχει τις εγγραφές όλων των γραμμών και από την στήλη 1 ως και την στήλη  $q$ , ενώ ο  $C_2$  τις εγγραφές όλων των γραμμών και από την στήλη  $q+1$  ως το τέλος.

Το αρχικό μοντέλο τάξης  $n$  μειώνεται σε ένα μικρότερο μοντέλο τάξης  $q$ , όπου  $q \ll n$ , διατηρώντας μόνο τις  $q$  πρώτες εγγραφές από τις συνολικές  $n$  του διανύσματος κατάστασης. Παράλληλα, οι πίνακες  $A$ ,  $B$ ,  $C$  αντικαθίστανται από τους επάνω αριστερά υποπίνακες  $A_{11}$ ,  $B_1$ ,  $C_1$  με μεγέθη  $q \times q$ ,  $q \times m$  και  $p \times q$  αντίστοιχα [6]. Με άλλα λόγια, το μειωμένο μοντέλο λαμβάνεται διαγράφοντας τις τελευταίες  $n-q$  γραμμές από τους πίνακες  $A$ ,  $B$  και τις τελευταίες  $n-q$  στήλες από τους πίνακες  $A$ ,  $C$ .

$$\hat{A} = \begin{bmatrix} A_{11} & \cancel{A_{12}} \\ \cancel{A_{21}} & \cancel{A_{22}} \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} B_1 \\ \cancel{B_2} \end{bmatrix}, \quad \hat{C} = [C_1 \quad \cancel{C_2}]$$

Ο πίνακας  $D$  παραμένει ίδιος με τον αρχικό, δηλαδή:

$$\hat{D} = D$$

Συνεπώς,

$$\hat{A} = A_{11}$$

$$\hat{B} = B_1$$

$$\hat{C} = C_1$$

$$\hat{D} = D$$

Τελικά, με το σύμβολο ' $\sim$ ' θα συμβολίσουμε τους νέους, μειωμένης διάστασης πίνακες, οι οποίοι επιστρέφονται ως αποτέλεσμα του αλγορίθμου TBR:

$$\tilde{A} = A_{11} \quad \tilde{B} = B_1$$

$$\tilde{C} = C_1 \quad \tilde{D} = D$$

Η μέθοδος TBR είναι ιδιαίτερα δημοφιλής εξαιτίας του ότι είναι υπολογιστικά απλή και παρέχει την δυνατότητα, όπως έχουμε αναφέρει και σε προηγούμενο εδάφιο, να επιλέξουμε την καινούρια τάξη συστήματος με την βοήθεια των HSV της κύριας διαγώνιου. Επιτυγχάνει προσεγγιστικά άριστα μοντέλα. Η μοναδική δυσκολία που διαφαίνεται είναι ότι στα πολύ μεγάλης κλίμακας συστήματα, η επίλυση των Lyapunov εξισώσεων θα έχει μεγάλο υπολογιστικό κόστος και θα είναι αρκετά χρονοβόρα. Ωστόσο, πρόσφατα αποτελέσματα ερευνών παρουσιάζουν προσεγγίσεις των δύο εξισώσεων, οι οποίες είναι χαμηλής τάξης και υπολογιστικά συμφέρουσες.

## 5.3 Παραλλαγή του TBR

---

**Αλγόριθμος 2:** DC – Accuracy Truncated Balanced Realization

---

*1: Εφάρμοσε τον μετασχηματισμό  $A' = A^{-1}$ ,  $B' = -A^{-1}B$ ,  $C' = CA^{-1}$ ,  $D' = D - CA^{-1}B$ .*

*2: Εφάρμοσε τον αλγόριθμο 1.*

*3: Εφάρμοσε τον αντίστροφο μετασχηματισμό  $\tilde{A}' = \tilde{A}^{-1}$ ,  $\tilde{B}' = -\tilde{A}^{-1}\tilde{B}$ ,  $\tilde{C}' = \tilde{C}\tilde{A}^{-1}$ ,  $\tilde{D}' = \tilde{D} - \tilde{C}\tilde{A}^{-1}\tilde{B}$ .*

---

Η βασική προσέγγιση της μεθόδου TBR που έχουμε παρουσιάσει μέχρι στιγμής, οδηγεί συχνά σε νέα μοντέλα συστήματος, των οποίων η συνάρτηση μεταφοράς δεν προσεγγίζει ικανοποιητικά την συνάρτηση μεταφοράς του αρχικού στις χαμηλές συχνότητες (dc frequency). Ο αλγόριθμος συμπεριφέρεται πολύ καλά στις υψηλές τιμές συχνοτήτων, ενώ στις dc συχνότητες έχει μερικές αδυναμίες. Για παράδειγμα, με είσοδο την κρουστική απόκριση παίρνουμε ως αποτέλεσμα δύο συναρτήσεις μεταφοράς που είναι προσεγγιστικά κοντά, ενώ για είσοδο την βηματική απόκριση η απόκλιση των δύο συναρτήσεων μεταφοράς είναι αρκετά μεγάλη με σφάλμα που τείνει στο άπειρο. Το παραπάνω αποτελεί μια σκοτεινή πτυχή του αλγορίθμου και για αυτό εν συνεχεία παρουσιάζουμε μια παραλλαγή του που θεραπεύει αυτή την αδυναμία.

Ως μηχανικοί, για να λύσουμε αυτό το πρόβλημα ένας τρόπος είναι ο μετασχηματισμός του αρχικού συστήματος σε ένα ‘αμοιβαίο σύστημα’. Η νέα προσέγγιση έγκειται στο γεγονός ότι μπορούμε να διαμερίσουμε το διάνυσμα κατάστασης σε 2 ομάδες μεταβλητών, στις ‘γρήγορες’ και στις ‘αργές’. Η πρώτη ομάδα, περιλαμβάνει τις καταστάσεις που γρήγορα συγκλίνουν σε σταθερές τιμές γύρω από την γειτονιά συγκεκριμένων συχνοτήτων, δηλαδή σε σταθερές τιμές γύρω από το  $s = 0$  για την περίπτωση των dc συχνοτήτων. Η δεύτερη ομάδα, συμπεριλαμβάνει τις υπόλοιπες καταστάσεις.

Πιο αναλυτικά, εάν θεωρήσουμε ένα μοντέλο συστήματος σε state-space μορφή και συνάρτηση μεταφοράς  $H(s)$ , το τροποποιημένο σύστημα σε state-space μορφή θα έχει συνάρτηση μεταφοράς  $H(s^{-1})$ . Ο μετασχηματισμός που θα κάνουμε αντιστοιχίζει το  $s = 0$  στο  $s = \infty$  και συνεπώς κατά την εξισορρόπηση (balancing) του συστήματος, παρουσιάζει καλύτερες προσεγγίσεις στις χαμηλές συχνότητες.

Επιπροσθέτως, τα όρια σφάλματος υπολογίζονται όπως ακριβώς και στην βασική εκδοχή TBR και οι όποιες ιδιότητες του αρχικού συστήματος μετά τον μετασχηματισμό εξακολουθούν να ισχύουν και στο νέο σύστημα.

#### Βήματα DC – Accurate TBR

Εφαρμόζουμε τον αμοιβαίο μετασχηματισμό στο αρχικό μοντέλο συστήματος  $(A, B, C, D)$  ως ακολούθως:

$$A' = A^{-1} \quad (5.16)$$

$$B' = -A^{-1} B \quad (5.17)$$

$$C' = C A^{-1} \quad (5.18)$$

$$D' = D - C A^{-1} B \quad (5.19)$$

Επομένως, το νέο μοντέλο συστήματος θα είναι  $(A', B', C', D')$  και το οποίο εισάγεται ως όρισμα στον αλγόριθμο TBR. Κατόπιν, τα βήματα του αλγορίθμου εκτελούνται όπως ακριβώς περιεγράφηκαν στο πρώτο μισό του τρέχοντος κεφαλαίου. Αμέσως πριν επιστρέψει ο TBR τα αποτελέσματά του, εκτελούμε στο μικρότερης διάστασης σύστημα  $(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$  τους εξής μετασχηματισμούς:

$$\tilde{A}' = \tilde{A}^{-1} \quad (5.20)$$

$$\tilde{B}' = -\tilde{A}^{-1} \tilde{B} \quad (5.21)$$

$$\tilde{C}' = \tilde{C} \tilde{A}^{-1} \quad (5.22)$$

$$\tilde{D}' = \tilde{D} - \tilde{C} \tilde{A}^{-1} \tilde{B} \quad (5.23)$$

Κατά συνέπεια, μέσω των μετασχηματισμών έχουμε επιτύχει καλή προσέγγιση των συναρτήσεων μεταφοράς και στις χαμηλές συχνότητες. Το νέο μοντέλο που επιστρέφει η παραλλαγή του TBR είναι το  $(\tilde{A}', \tilde{B}', \tilde{C}', \tilde{D}')$ .

Εν κατακλείδι, οι συναρτήσεις μεταφοράς που βασίζονται στην μεταβλητή  $s$  τείνουν να ταιριάζουν τέλεια με τις αντίστοιχες αρχικές στο άπειρο, ενώ όσες βασίζονται στην μεταβλητή  $s^{-1}$  τείνουν να ταιριάζουν άψογα στις DC συχνότητες.

Η σελίδα αυτή είναι σκόπιμα λευκή.

# ΚΕΦΑΛΑΙΟ 6

## Υλοποίηση της Προσομοίωσης

Στο τρέχον κεφάλαιο εστιάζουμε κατά κύριο λόγο στο τι είναι μια προσομοίωση, τι χρειάζεται για να γίνει και πως γίνεται να μοντελοποιηθεί ένα κύκλωμα που δίνεται μέσα από ένα αρχείο περιγραφής κυκλώματος. Ακολούθως, περιγράφουμε πως ενεργοποιείται η μέθοδος υποβιβασμού τάξης TBR.

### 6.1 Αρχείο περιγραφής κυκλώματος

Η υλοποίησή μας ξεκινάει από το σημείο όπου λαμβάνουμε ένα αρχείο ή ένα σύνολο από αρχεία, τα οποία περιγράφουν τη κυκλωματική συνδεσμολογία και τοπολογία με κάποιο συγκεκριμένο πρότυπο, το οποίο είναι κοινό για τις περισσότερες εφαρμογές προσομοίωσης κυκλωμάτων.

Τα αρχεία περιγραφής κυκλώματος ή αλλιώς netlists είναι αρχεία σε μορφή κειμένου (.txt), τα οποία οργανώνονται σε διαφορετικά τμήματα. Το τμήμα εκείνο που θα μας απασχολήσει περισσότερο στην παρούσα μελέτη είναι η λίστα με την περιγραφή των κυκλωματικών στοιχείων, η οποία περιέχει όλη την πληροφορία σχετικά με τα χαρακτηριστικά των διαφόρων στοιχείων αλλά και το πώς αυτά συνδέονται μεταξύ τους. Τα τμήματα, τα οποία ακολουθούνται από το σύμβολο «%» αποτελούν σχόλια και αγνοούνται.

Ειδικότερα, τα αποτελέσματα που λαμβάνουμε με επισκόπηση της τοπολογίας του κυκλώματος γράφονται σε ένα αρχείο, το οποίο για την περίπτωση μας είναι σε MatrixMarket (MM) format [15]. Σε αυτήν την κατηγορία, μπορούμε να συναντήσουμε μία από τις εξής 2 υποκατηγορίες μορφοποίησης, το Coordinate Format και το Array Format, που δηλώνονται στο header line του αρχείου. Η πρώτη μορφή είναι κατάλληλη για την απεικόνιση αραιών πινάκων. Στο αρχείο αναπαρίστανται μόνο οι μη μηδενικές εγγραφές καθώς και ο αριθμός γραμμής και στήλης που εντοπίστηκε αυτό το μη μηδενικό στοιχείο. Η δεύτερη μορφή είναι κατάλληλη για την αναπαράσταση πυκνών πινάκων.

Επιπροσθέτως, στην εισαγωγική γραμμή (header line) του αρχείου μπορεί να περιέχονται χαρακτηριστικές λέξεις - κλειδιά, όπως real, complex, integer, pattern, οι

οποίες υποδηλώνουν τον τύπο των αριθμητικών τιμών των εγγραφών που πρόκειται να διαβαστούν. Με τον όρο `pattern` δεν δηλώνουμε κάποιον τύπο, παρά μόνο τη θέση της μη μηδενικής εγγραφής. Ακόμη, με τις λέξεις - κλειδιά `general`, `symmetric`, `skew-symmetric`, `Hermitian`, υποδεικνύεται ο τρόπος που θα αποθηκευτούν οι εγγραφές. Για παράδειγμα, μια συμμετρική μορφή πίνακα, ενδεχομένως να μειώσει το μέγεθος του αρχείου δεδομένων, αποθηκεύοντας τις μη μηδενικές εγγραφές πάνω ή κάτω από την κύρια διαγώνιο του πίνακα. Στην `skew-symmetric` τα στοιχεία της διαγώνιου είναι μηδέν και συνεπώς παραλείπονται. Παρατίθεται ένα ενδεικτικό παράδειγμα μορφοποίησης αρχείου:

```
%%MatrixMarket matrix coordinate real general    <--- header line

%% comments

%   M   N   Value
%   I1 J1  A(I1, J1)
%   I2 J2  A(I2, J2)
%   I3 J3  A(I3, J3)
%           . . .
%   IL JL  A(IL, JL)
```

Σε αυτό το αρχείο δηλώνεται ότι θα έχουμε έναν αραιό πίνακα, οι τιμές σε κάθε κελί του θα είναι πραγματικοί αριθμοί και θα αναπαρίσταται σε `general` μορφή, δηλαδή δεν θα λάβουμε υπόψιν συμμετρίες. Λόγου χάριν, εάν επιθυμούμε να πάρουμε ως αποτέλεσμα τον ακόλουθο 5 x 5 αραιό πίνακα (Πίνακας 6.1):

Στήλη1	Στήλη2	Στήλη3	Στήλη4	Στήλη5
1.77	0	0	0	0
0	1.92	0	0	0
0	0	1.35	0	0
0	0	0	1.29	0
0	0	0	0	1.21

**Πίνακας 6.1:** Αποτέλεσμα ανάγνωσης αρχείου

σε `MatrixMarket Coordinate real general format` το αρχείο θα έχει την ακόλουθη μορφή (Πίνακας 6.2):

Στήλη1	Στήλη2	Στήλη3
1	1	1,77E-06
2	2	1,92E-06
3	3	1,21E-06
4	4	1,29E-06
5	5	1,21E-06

**Πίνακας 6.2:** *MatrixMarket* φορμάτ αρχείου

Τα δύο αρχεία που μας δόθηκαν για την προσομοίωση του αλγορίθμου TBR βρίσκονται σε MatrixMarket matrix Coordinate real general format. Αναφέρονται στις αγωγιμότητες και χωρητικότητες ενδεικτικού κυκλώματος και με την συνάρτηση MatrixMarketRead.m θα παράγουμε τους αντίστοιχους πίνακες. Στην συνέχεια, με δεδομένους αυτούς τους πίνακες θα δημιουργήσουμε τις MNA εξισώσεις του κυκλώματος.

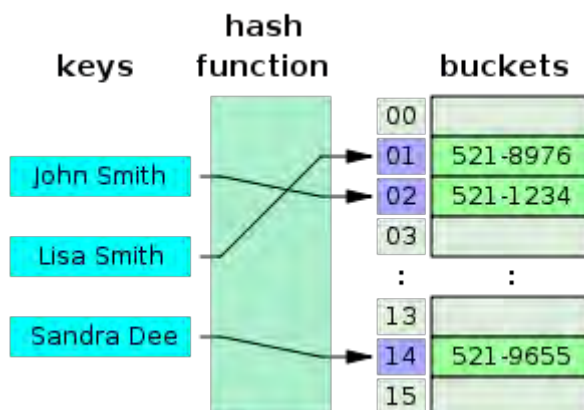
## 6.2 Κατασκευή Προσομοιωτή

Ο προσομοιωτής (parser) είναι ένα πρόγραμμα που υλοποιεί μια διαδικασία ανάλυσης του κειμένου που περιέχεται στο αρχείο περιγραφής του κυκλώματος. Αναλύει ουσιαστικά μια αλληλουχία συμβόλων και με βάση ένα σύνολο κανόνων αποφασίζει ποια από αυτά θα αγνοήσει και ποια είναι εκείνα που περιέχουν χρήσιμη πληροφορία, έτσι ώστε να τη συλλέξει και να την αποθηκεύσει σε κατάλληλες δομές δεδομένων. Ο parser, αφού ολοκληρώσει την επεξεργασία του αρχείου περιγραφής, παράγει ως έξοδο δομές δεδομένων, σε μορφή πίνακα ή λιστών.

Αναλυτικότερα, το πρώτο στάδιο για τη δημιουργία ενός προσομοιωτή είναι το διάβασμα ενός αρχείου το οποίο όπως προαναφέραμε περιγράφει την τοπολογία και την συνδεσμολογία του κυκλώματος προς εξέταση. Αυτή την πληροφορία που διαβάζουμε από το αρχείο θα πρέπει να τη μεταφέρουμε στον προσομοιωτή μας. Για αυτό το λόγο διαβάζουμε το αρχείο γραμμή προς γραμμή και στη συνέχεια όλες τις παραμέτρους του κάθε στοιχείου.

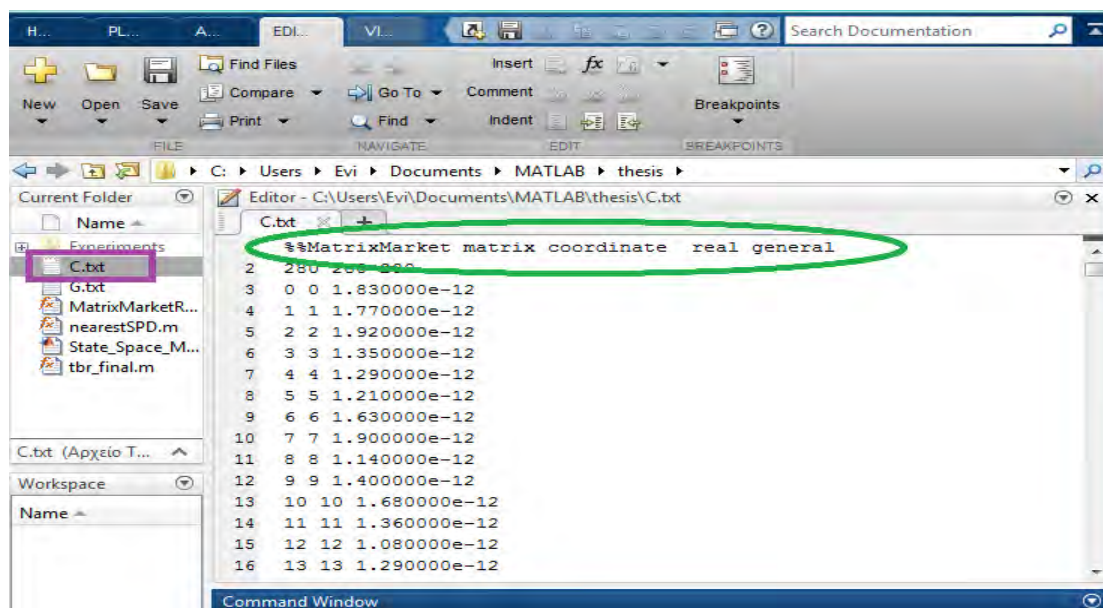
Είναι χρήσιμο λοιπόν, καθώς διαβάζουμε το αρχείο, να αποθηκεύουμε όλη αυτή την πληροφορία σε κάποια κατάλληλη δομή δεδομένων, ώστε να μη χρειάζεται να χρησιμοποιούμε συνεχώς το αρχείο σε ό,τι κάνουμε στη συνέχεια. Οι πίνακες θεωρούνται καλή δομή με αξιόλογες ιδιότητες. Θα μπορούσαμε να χρησιμοποιήσουμε για αυτόν τον σκοπό για παράδειγμα, τον πίνακα κατακερματισμού (hash table). Όπως γνωρίζουμε από την θεωρία, ο πίνακας κατακερματισμού (Εικόνα 21) έχει πολλά

πλεονεκτήματα σαν δομή αποθήκευσης δεδομένων με τα βασικά του να είναι η επίδοση στο χώρο που καταναλώνει αλλά και η ταχύτητα ανάγνωσης των δεδομένων.



Εικόνα 21: Παράδειγμα τηλεφωνικού καταλόγου σε μορφή hash table

Στην παρούσα διπλωματική εργασία, δημιουργήσαμε σε MATLAB την συνάρτηση MatrixMarketRead.m, η οποία αποθηκεύει τις εγγραφές του αρχείου σε πίνακα κατακερματισμού, είτε αραιό είτε πυκνό, ανάλογα με το φορμάτ που υποδεικνύεται στην επικεφαλίδα του αρχείου (Εικόνα 22). Παράλληλα, προαιρετικά στην έναρξη του αρχείου δηλώνεται ο αριθμός των γραμμών και των στηλών του πίνακα που θα δημιουργηθεί καθώς και ο αριθμός των μη μηδενικών εγγραφών του. Διαβάζουμε από τα αρχεία C.txt και G.txt τις πληροφορίες για τις χωρητικότητες και τις διαγωγιμότητες του κυκλώματος και δημιουργούμε τους πίνακες 'C' και 'G' αντίστοιχα.



Εικόνα 22: MatrixMarket Format αρχείου



## 6.3 MNA σύστημα του κυκλώματος

Έπειτα και αφού βεβαιωθούμε πως όλα κύλησαν ομαλά με το διάβασμα των δύο αρχείων και την αποθήκευση της πληροφορίας τους στη δομή δεδομένων μας, έχουμε να μοντελοποιήσουμε την πληροφορία, κατασκευάζοντας ουσιαστικά με βάση τους πίνακες το σύστημα MNA του κυκλώματος. Ο τρόπος εξαγωγής των MNA εξισώσεων περιεγράφηκε αναλυτικά στο κεφάλαιο 2 και έτσι εδώ δεν θα προχωρήσουμε σε περαιτέρω λεπτομέρειες. Συνοπτικά, το σύστημα είναι στην ακόλουθη MNA μορφή:

$$G\underline{x}(t) + C\dot{\underline{x}}(t) = B\underline{u}_{in}(t)$$

$$V_{out}(t) = L^T \underline{x}(t)$$

Ισοδύναμα, σε state-space μορφή θα είναι:

$$\dot{\underline{x}}(t) = A \underline{x}(t) + B \underline{u}(t)$$

$$V_{out}(t) = C^T \underline{x}(t)$$

Η βασική ιδέα είναι πως κάθε στοιχείο που διαβάζεται από το αρχείο και αποθηκεύεται στη δομή δεδομένων, ανήκει και σε μια συγκεκριμένη ομάδα στοιχείων που επηρεάζουν ανάλογα τη συμπεριφορά του κυκλώματος. Αυτές οι ομάδες είναι δύο. Η ομάδα 1, η οποία περιέχει αντιστάσεις, πυκνωτές και πηγές ρεύματος και η ομάδα 2, η οποία περιέχει πηνία και πηγές τάσης. Οι εξισώσεις των στοιχείων της πρώτης ομάδας μπορούν να γραφούν, έστω για κάποιο στοιχείο  $k$ , στην μορφή:

$$i_k(t) = g_k u_k(t) + C_k \frac{du_k}{dt} + s_k(t) \quad (6.1)$$

όπου  $s_k(t)$  γνωστή συνάρτηση ως προς  $t$ . Τα στοιχεία των οποίων οι εξισώσεις δεν μπορούν να γραφούν στην μορφή (6.1) ή επιζητείται το ρεύμα τους, ανήκουν στην δεύτερη ομάδα. Αν συμβολίσουμε με  $m_1$  τα στοιχεία της πρώτης ομάδας και με  $m_2$  τα στοιχεία της δεύτερης ομάδας, τότε:  $m_1 + m_2 = m$ , όπου  $m$  ο συνολικός αριθμός στοιχείων ή καταστάσεων του κυκλώματος.

Αφού διαχωρίσουμε λοιπόν το σύνολο των στοιχείων  $m$ , στις ομάδες  $m_1$  και  $m_2$ , είμαστε σε θέση να χωρίσουμε αντίστοιχα τον πίνακα και τα διανύσματα ρευμάτων και τάσεων επίσης σε 2 υποομάδες. Χρησιμοποιούμε τους νόμους του Kirchhoff για τα ρεύματα και τις τάσεις (KCL, KVL), ώστε να εξάγουμε τις εξισώσεις και εν συνεχεία να ανάγουμε το σύστημα εξισώσεων σε μορφή πίνακα. Με τη βοήθεια αυτών των πινάκων καταλήγουμε στο MNA σύστημα του κυκλώματος μας.

### 6.4 Χρήση μεθόδων υποβιβασμού τάξης μοντέλου

Αφού μοντελοποιήσουμε λοιπόν το κύκλωμα που μας δόθηκε από το αρχικό αρχείο και εξάγουμε το MNA σύστημα του κυκλώματος με τις τεχνικές που περιγράψαμε στο κεφάλαιο 2, είμαστε σε θέση πλέον να χρησιμοποιήσουμε κάποια μέθοδο υποβιβασμού τάξης μοντέλου προκειμένου να απλοποιήσουμε αυτό το σύστημα. Ως εκ τούτου, το σύστημα θα λυθεί πολύ γρηγορότερα και αποδοτικότερα σε σχέση με το αρχικό, χωρίς όμως να υστερεί σε ακρίβεια η λύση. Η προσομοίωση θα γίνει πολύ γρηγορότερα.

Στην τρέχουσα διπλωματική εργασία η μέθοδος MOR που μελετήθηκε αλλά και υλοποιήθηκε είναι η TBR. Ο αλγόριθμος της αναλύθηκε εκτενώς στο κεφάλαιο 5. Η TBR θα λάβει ως αρχικό όρισμα τους πίνακες  $A$ ,  $B$ ,  $C$  και  $D$  και θα επιστρέψει τους καινούριους, μειωμένης διάστασης πίνακες  $\tilde{A}$ ,  $\tilde{B}$ ,  $\tilde{C}$  και  $\tilde{D}$ . Με δεδομένες αυτές τις εισόδους λοιπόν, εφαρμόζεται ο αλγόριθμος του κεφαλαίου 5 προκειμένου να φτάσουμε στο επιθυμητό αποτέλεσμα, μέσα από τη δημιουργία της συνάρτησης μεταφοράς του αρχικού κυκλώματος και της συνάρτησης μεταφοράς που προκύπτει μετά τη χρήση της TBR. Ως απόρροια, οι δύο συναρτήσεις μεταφοράς σχεδόν θα ταυτίζονται, πράγμα που επιθυμούμε.

#### *Επιπρόσθετες παρατηρήσεις*

Από τη νέα αυτή συνάρτηση μεταφοράς μπορεί να δημιουργηθεί ένα νέο ισοδύναμο κύκλωμα για το οποίο όμως απαιτούνται πολύ λιγότερες απαιτήσεις προκειμένου να επιλυθεί. Είναι απολύτως βέβαιο πως το νέο συρρικνωμένο σύστημα θα καταναλώνει λιγότερη μνήμη και επεξεργαστική ισχύ, ενώ ταυτόχρονα θα λυθεί γρηγορότερα συγκριτικά με το αρχικό.

Η σελίδα αυτή είναι σκόπιμα λευκή.

# ΚΕΦΑΛΑΙΟ 7

## Εκτίμηση της Λύσης - Αποτελέσματα – Συγκρίσεις

Στο παρόν κεφάλαιο παρουσιάζονται τα αρχεία περιγραφής κυκλωμάτων της IBM που χρησιμοποιήθηκαν για την διεξαγωγή των πειραμάτων και την αξιολόγηση του αλγορίθμου TBR, η πειραματική διαδικασία με τις γραφικές παραστάσεις αρχικού και μειωμένου μοντέλου, στατιστικά στοιχεία καθώς και εκτίμηση των αποτελεσμάτων.

### 7.1 Πειραματικά Κυκλώματα

Οι κώδικες καθώς και οι προσομοιώσεις για τις ανάγκες της παρούσας διπλωματικής εργασίας, όπως αναφέραμε ξανά, έγιναν στο γραφικό περιβάλλον της MATLAB με χρήση του “Control System Toolbox”. Όλες οι εκτελέσεις πραγματοποιήθηκαν με την βοήθεια H/Y Acer Aspire 5750G με τα ακόλουθα τεχνικά χαρακτηριστικά που φαίνονται στον πίνακα 7.1:

<b>Operating System Version</b>	Microsoft Windows 7 Home Premium 2009 Service Pack 1
<b>Processor</b>	Intel(R) Core(TM) i5 – 2410M CPU @ 2.30GHz 2.30 GHz
<b>Processor Max Speed</b>	3000 MHz
<b>Total Physical Memory</b>	5,85 GB

Πίνακας 7.1: Χαρακτηριστικά του συστήματος μας

Στον πίνακα 7.2 παρουσιάζονται τα χαρακτηριστικά ενός ενδεικτικού κυκλώματος που χρησιμοποιήθηκε για τον έλεγχο της υλοποίησής μας. Αρχικά, δοκιμάσαμε με πίνακες σχετικά μικρής διάστασης, έτσι ώστε να μπορούμε ευκολότερα να αποφανθούμε και να ελέγξουμε την ορθότητα των αποτελεσμάτων. Ενδεικτικά, οι πίνακες εισόδου C και G είναι διάστασης 280 x 280. Οι μη-μηδενικές εγγραφές του πίνακα C είναι 280, ενώ του πίνακα G, 1000.

Name	Nodes	Current Sources	Resistances	Capacitors
IBM	281	4	360	280

**Πίνακας 7.2:** Ενδεικτικό Benchmark μικρής διάστασης πίνακα

## 7.2 Πειραματική Διαδικασία

Στο συγκεκριμένο εδάφιο παρουσιάζονται τα αποτελέσματα που πήραμε από την προσομοίωση της μεθόδου TBR που υλοποιήθηκε σε MATLAB. Για την δεδομένη μέθοδο έγινε η προσομοίωση του κυκλώματος (πίνακας 7.2) χωρίς ελαχιστοποίηση και στην συνέχεια με ελαχιστοποίηση. Το ενδεικτικό κύκλωμα (πίνακας 7.2) ελαχιστοποιείται μέσω της συνάρτησης `minreal` στις 249 καταστάσεις και μετά από εφαρμογή της TBR σε 224 καταστάσεις. Ειδικότερα, στην παρούσα υπό ενότητα θα αναφερθούμε στα ακόλουθα:

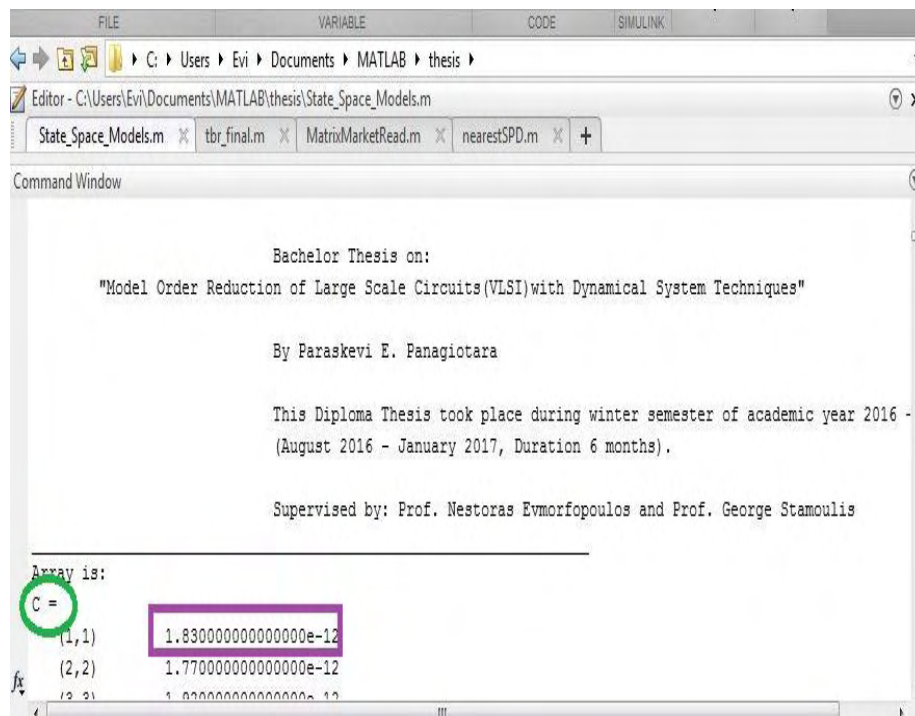
- Ανάλυση υλοποίησης συναρτήσεων διπλωματικής εργασίας (`State_Space_Models.m`, `tbr_final.m`, `nearestSPD.m`, `MatrixMarketRead.m`).
- Αποτελέσματα της αρχικής και της τελικής συνάρτησης μεταφοράς, μετά την εφαρμογή αλγορίθμου TBR, με την βοήθεια διαγραμμάτων Bode. Το διάγραμμα Bode είναι ένα από τα ισχυρότερα εργαλεία ανάλυσης και σχεδίασης συστημάτων ελέγχου σε περιβάλλον γραφικών.
- Χρόνοι που χρειάστηκαν για την παραγωγή των υπό πινάκων και την προσομοίωση καθώς και αξιολόγηση της κατανάλωσης πόρων συστήματος.

### 7.2.1 Ανάλυση υλοποίησης συναρτήσεων διπλωματικής εργασίας

Προτού προχωρήσουμε στην έκθεση των γραφικών παραστάσεων θα παρουσιάσουμε ορισμένες παρατηρήσεις αναφορικά με τους κώδικες υλοποίησης. Επισημαίνουμε πως ισχύει,  $s = 2\pi f$ . Θα πρέπει να γίνει κάποια κατάλληλη επιλογή συχνότητας, έστω  $f = 1e8$ ,

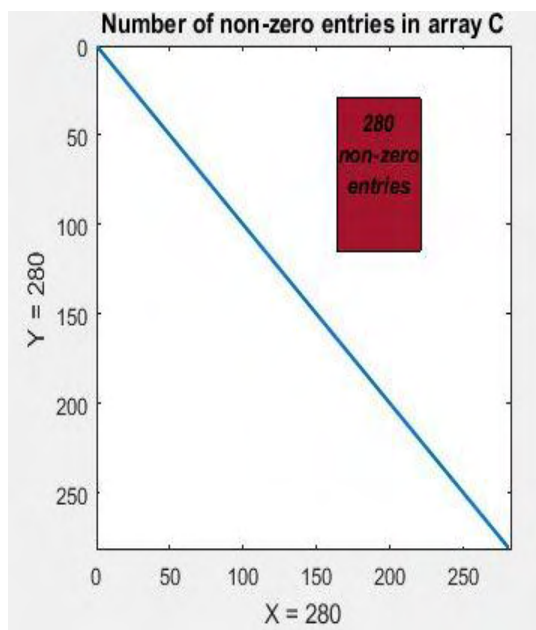
ώστε να γίνει η δημιουργία των πινάκων (π.χ.  $G+sC$ ). Χρειάζεται η κατάλληλη αρχικοποίηση ώστε οι πίνακες που προκύπτουν να μην είναι singular και κατά συνέπεια να ορίζεται ο πίνακας  $C^{-1}$ .

Ξεκινώντας, αναλύουμε το περιεχόμενο του script 'State\_Space\_Models.m'. Δίνουμε ως είσοδο στην συνάρτηση MatrixMarketRead.m το αρχείο εισόδου C.txt. Η συνάρτηση επιστρέφει ως αποτέλεσμα το περιεχόμενο του αρχείου σε μορφή πίνακα. Μάλιστα, για να ορίζεται ο πίνακας  $C^{-1}$  και να μην υπάρχουν άπειρες τιμές στα στοιχεία της διαγώνιου, θέτουμε σε αυτά στον πίνακα C, την προκαθορισμένη τιμή  $2.5 \cdot 10^{-12}$ . Για τις μετρήσεις χρησιμοποιήθηκε ακρίβεια 15 δεκαδικών ψηφίων, ώστε να προσδιορίσουμε ποια θα είναι η προσέγγιση την οποία θέλουμε να επιτύχουμε και να είμαστε ικανοποιημένοι με τη λύση. Πιο συγκεκριμένα, θα είναι (Εικόνα 23):

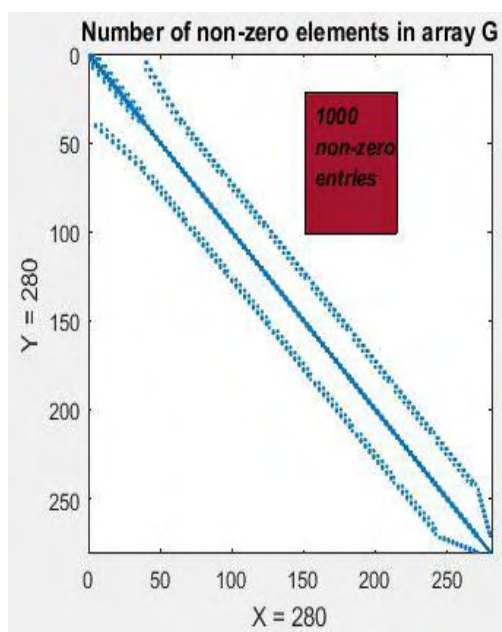


Εικόνα 23: Απεικόνιση πίνακα C

Όμοια, δημιουργείται από το αρχείο G.txt και ο πίνακας G. Επίσης, στο σημείο αυτό χρησιμοποιήσαμε την εντολή 'spy', ώστε να οπτικοποιήσουμε τις μη μηδενικές εγγραφές των πινάκων C,G, οι οποίες είναι 280 και 1000 αντίστοιχα (Εικόνες 24, 25). Όλες οι εγγραφές του πίνακα C βρίσκονται επάνω στην διαγώνιο (ευθεία γραμμή), ενώ για τον πίνακα G εντοπίζουμε εγγραφές και εκτός της κύριας διαγώνιου.



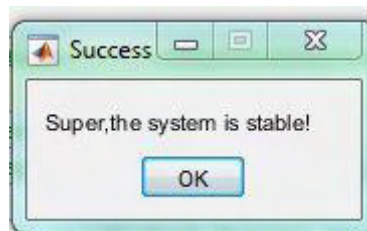
Εικόνα 24: Μη μηδενικές εγγραφές πίνακα C



Εικόνα 25: Μη μηδενικές εγγραφές πίνακα G

Στην συνέχεια, στο παράθυρο εντολών εμφανίζεται προτροπή προς τον χρήστη, ώστε να επιλέξει τον τρόπο εισαγωγής του συστήματος. Επιλέγει 1, αν επιθυμεί να εισάγει το σύστημα σε μορφή state-space, ή επιλέγει 2, αν επιθυμεί να το δώσει σε μορφή συνάρτησης μεταφοράς. Στην περίπτωση που επιλέξει 1, προτρέπεται να εισάγει τους πίνακες A,B,C,D, με τον τρόπο που έχει οριστεί παραπάνω. Μετά από την δημιουργία του πίνακα  $A = -C^{-1}G$ , ελέγχεται εάν όλες οι ιδιοτιμές του πίνακα, είναι μη αρνητικές. Στην περίπτωση όπου έστω και μια τιμή είναι μη αρνητική, βλέπουμε ένα pop-up

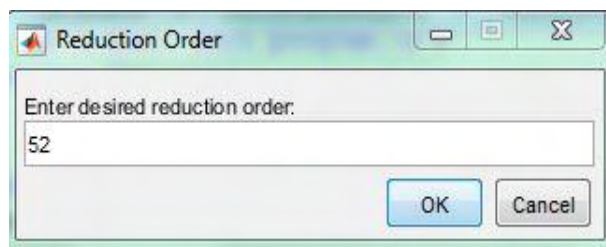
παράθυρο, το οποίο μας ενημερώνει ότι το σύστημα δεν είναι ευσταθές και θα πρέπει να δοκιμάσουμε με καινούριο σύστημα. Διαφορετικά, παίρνουμε μήνυμα ότι το σύστημα είναι ευσταθές (Εικόνα 26).



Εικόνα 26: Pop-up παράθυρο μηνύματος για ευστάθεια συστήματος

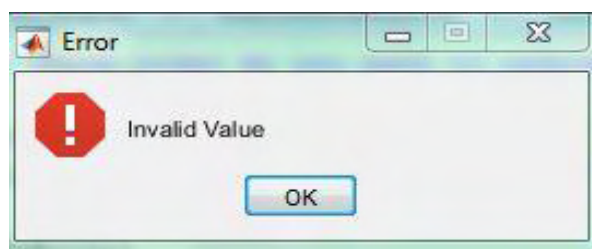
Κατόπιν, έχοντας δημιουργήσει το σύστημα, καλούμε την συνάρτηση `minreal`, έτσι ώστε να εξαλείψουμε τους κοινούς πόλους και μηδενικά από τον παρονομαστή και τον αριθμητή αντίστοιχα και κατά συνέπεια να μειωθεί ο αριθμός των καταστάσεων του συστήματος. Ελέγχουμε κατά πόσο ο πίνακας  $A$  είναι τετραγωνικός και εάν ο αριθμός των γραμμών του διανύσματος εισόδου είναι ίσος με την διάσταση του πίνακα  $A$ . Όμοια, ελέγχουμε κατά πόσο ο αριθμός στηλών του διανύσματος εξόδου είναι ίσος με την διάσταση του πίνακα  $A$ . Σε κάθε περίπτωση κατάλληλα μηνύματα λάθους εμφανίζονται.

Εν συνεχεία, ο χρήστης δίνει από το πληκτρολόγιο την επιθυμητή τάξη μείωσης του μοντέλου συστήματος (Εικόνα 27).



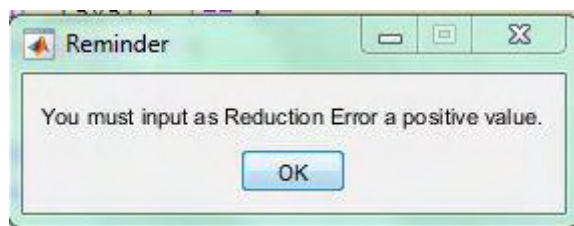
Εικόνα 27: Παράθυρο εισαγωγής νέας τάξης μοντέλου συστήματος

Στην περίπτωση όπου καταχωρήσει αρνητική τιμή ή μηδέν, εκτυπώνεται κατάλληλο μήνυμα λάθους (Εικόνες 28, 29) και εμφανίζεται ξανά το παράθυρο της εικόνας 27.



Εικόνα 28: Μη έγκυρη τάξη μοντέλου συστήματος





Εικόνα 29: Υπενθύμιση θετικής τιμής του Reduction Error

Έπειτα, στο παράθυρο εντολών εμφανίζεται προτροπή προς τον χρήστη ώστε να επιλέξει ποιος αλγόριθμος επιθυμεί να τρέξει. Επιλέγει 1, για την κλασσική παραδοχή του αλγορίθμου TBR, ενώ διαλέγει 2 αν επιθυμεί να τρέξει ο αλγόριθμος TBR αλλά με έμφαση στις DC συχνότητες. Κατόπιν, δημιουργούμε ένα struct με πεδία, το είδος του αλγορίθμου, το άνω όριο σφάλματος (ορίζεται στατικά) και την ανοχή των Hankel Singular Values. Υπάρχουν 4 διαφορετικοί τρόποι κλήσης της συνάρτησης υποβιβασμού τάξης μοντέλου, με βάση τα ορίσματα:

- ✓ Διαβάζουμε την νέα, μειωμένη τάξη συστήματος από το πληκτρολόγιο εντός της συνάρτησης TBR ή εντός της τρέχουσας συνάρτησης. Ενδεχομένως, να μην υπάρχουν τόσο καλές προσεγγίσεις στο αρχικό και στο τελικό μοντέλο, διότι δεν έχει οριστεί άνω όριο σφάλματος μεταξύ των δύο συναρτήσεων μεταφοράς.

```
[rsys1,q,TBalInverse,TBal,HSV] = tbr_final(sys1)
[rsys1,q,TBalInverse,TBal,HSV] = tbr_final(sys1,q)
```

- ✓ Η νέα, μειωμένη τάξη συστήματος υπολογίζεται με βάση το άνω όριο σφάλματος, το οποίο για την περίπτωση μας έχει οριστεί στατικά μέσω του struct s. Το συγκεκριμένο όριο σφάλματος αποδείχθηκε, έπειτα από δοκιμές που πραγματοποιήσαμε, ότι δίνει τις καλύτερες προσεγγίσεις μεταξύ αρχικής και τελικής συνάρτησης μεταφοράς.

```
[rsys1,q,TBalInverse,TBal,HSV] = tbr_final(sys1,s)
```

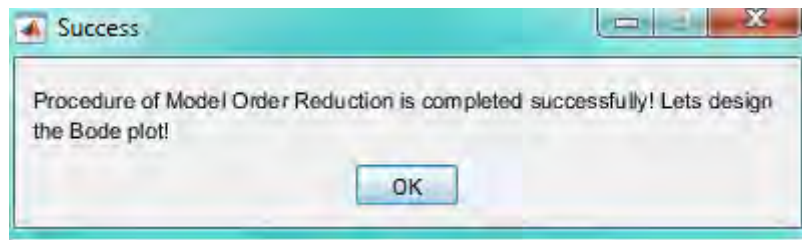
- ✓ Εισάγουμε την νέα τάξη μοντέλου συστήματος και το όριο σφάλματος ως όρισμα της συνάρτησης. Ενδεχομένως, να μην υπάρχουν τόσο καλές προσεγγίσεις στο αρχικό και στο τελικό μοντέλο, διότι το άνω όριο σφάλματος δεν λαμβάνεται υπόψιν τελικά στην συνάρτηση TBR.

```
[rsys1,q,TBalInverse,TBal,HSV] = tbr_final(sys1,q,s)
```

Στο σημείο αυτό καλείται η συνάρτηση με το όνομα `tbr_final` που βρίσκεται στο αρχείο `tbr_final.m`. Στο εσωτερικό της πραγματοποιείται η διαδικασία υποβιβασμού τάξης μοντέλου συστήματος ακολουθώντας ρητά τα βήματα που περιγράψαμε στο κεφάλαιο 5. Μάλιστα, μέσα από την συνάρτηση `tbr_final` καλούμε την συνάρτηση `nearestSPD` (βρίσκεται στο αρχείο `nearestSPD.m`) προκειμένου να

εξασφαλίσουμε ότι οι ιδιοτιμές της λύσης των εξισώσεων Lyapunov θα είναι θετικές και συνεπώς θα μπορεί να γίνει ανάλυση κατά Cholesky.

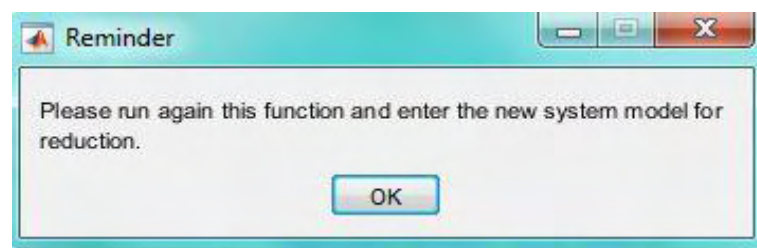
Όταν επιστρέφει η συνάρτηση TBR, εμφανίζεται κατάλληλο dialog box (Εικόνα 30) που δηλώνει τον τερματισμό της διαδικασίας υποβιβασμού τάξης μοντέλου μέσω της TBR και πατώντας 'OK' ξεκινάει η διαδικασία του σχεδιασμού γραφικών παραστάσεων του αρχικού και τελικού, μειωμένου μοντέλου:



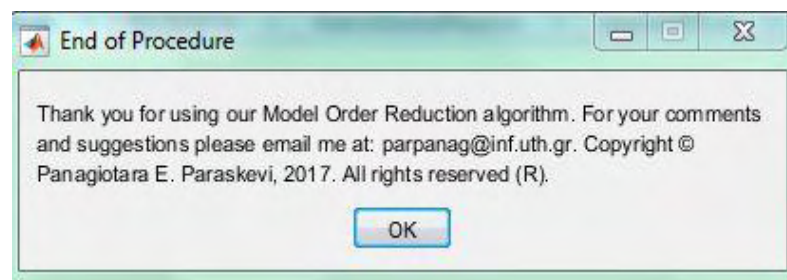
Εικόνα 30: Επιτυχής ολοκλήρωση της μεθόδου TBR

Παράλληλα, υπολογίζουμε την άπειρη νόρμα της διαφοράς του αρχικού και του τελικού συστήματος και επιβεβαιώνουμε πως έχει μικρή τιμή. Αυτό από άποψη διαγράμματος σημαίνει ότι οι δύο γραφικές παραστάσεις έχουν πολύ μικρή απόκλιση μεταξύ τους.

Στο τέλος, στο παράθυρο εντολών ο χρήστης προτρέπεται αν επιθυμεί να συνεχίσει με καινούριο αρχείο περιγραφής κυκλώματος ή να τερματίσει την διαδικασία. Στην πρώτη περίπτωση, επιλέγοντας 1, ενθαρρύνεται να τρέξει ξανά την ίδια συνάρτηση (Εικόνα 31), ενώ στην δεύτερη περίπτωση, επιλέγοντας 2, τερματίζει η συνάρτηση με αντίστοιχο μήνυμα κατοχής δικαιωμάτων (Εικόνα 32):



Εικόνα 31: Επανάληψη διαδικασίας



Εικόνα 32: Ολοκλήρωση διαδικασίας

## 7.2.2 Ενδεικτικά Αποτελέσματα Προσομοίωσης

Υποθέτουμε πως το μέγεθος των πινάκων C, G είναι size x size και το μέγεθος του πίνακα B είναι size x size\_B. Η δημιουργία των νέων πινάκων θα παραγάγει πίνακες με τις εξής διαστάσεις:

$$A \rightarrow (\text{size} \times \text{size}) \times (\text{size} \times \text{size}) \rightarrow \text{size} \times \text{size}$$

$$R \rightarrow (\text{size} \times \text{size}) \times (\text{size} \times \text{size\_B}) \rightarrow \text{size} \times \text{size\_B}$$

$$\tilde{G} \rightarrow \text{new\_size} \times \text{new\_size}$$

$$\tilde{C} \rightarrow \text{new\_size} \times \text{new\_size}$$

$$\tilde{B} \rightarrow \text{new\_size} \times \text{size\_B}$$

Τελικά, θα λάβουμε συνάρτηση μεταφοράς που θα έχει τις παρακάτω διαστάσεις:

$$\tilde{Z} = \tilde{B}^T (\tilde{G} + s\tilde{C})^{-1} \tilde{B}$$

$$\tilde{Z} \rightarrow (\text{size\_B} \times \text{new\_size}) \times (\text{new\_size} \times \text{new\_size}) \times (\text{new\_size} \times \text{size\_B})$$

$$\tilde{Z} \rightarrow (\text{size\_B} \times \text{size\_B}) \quad (7.1)$$

Κάνοντας αντίστοιχη διαδικασία και για τη διάσταση της συνάρτησης μεταφοράς του αρχικού κυκλώματος έχουμε:

$$Z = B^T (G + sC)^{-1} B$$

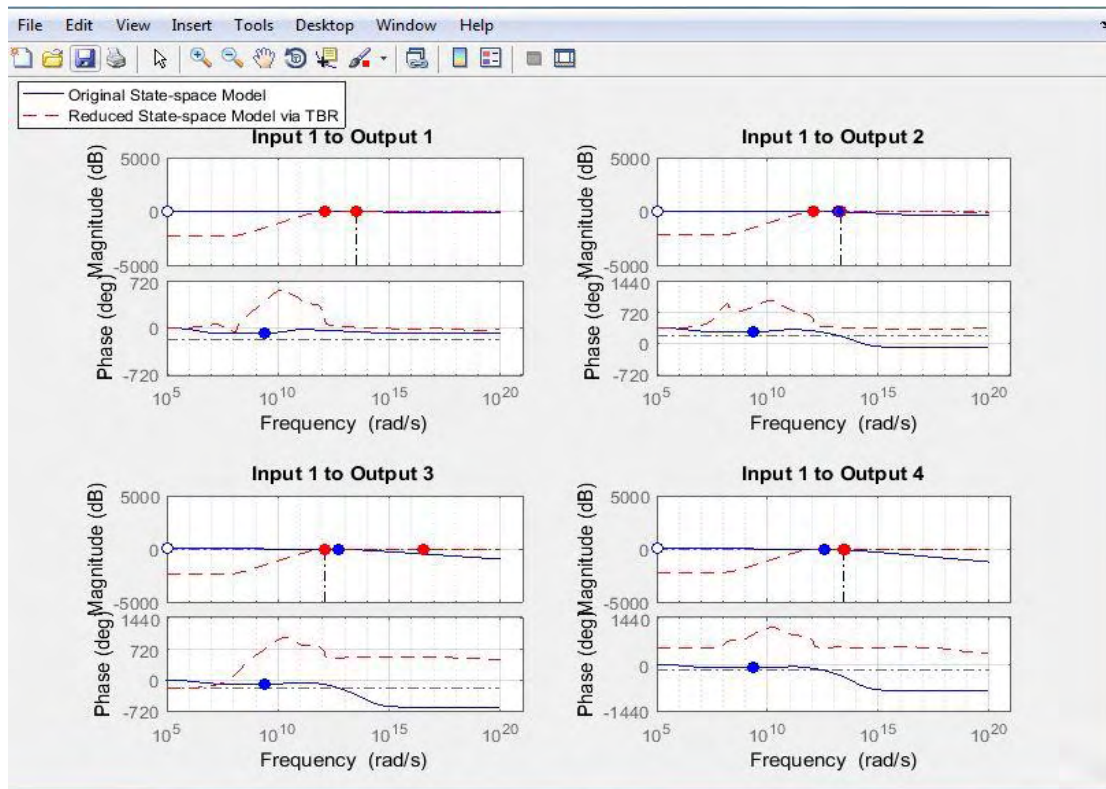
$$Z \rightarrow (\text{size\_B} \times \text{size}) \times (\text{size} \times \text{size}) \times (\text{size} \times \text{size\_B})$$

$$Z \rightarrow (\text{size\_B} \times \text{size\_B}) \quad (7.2)$$

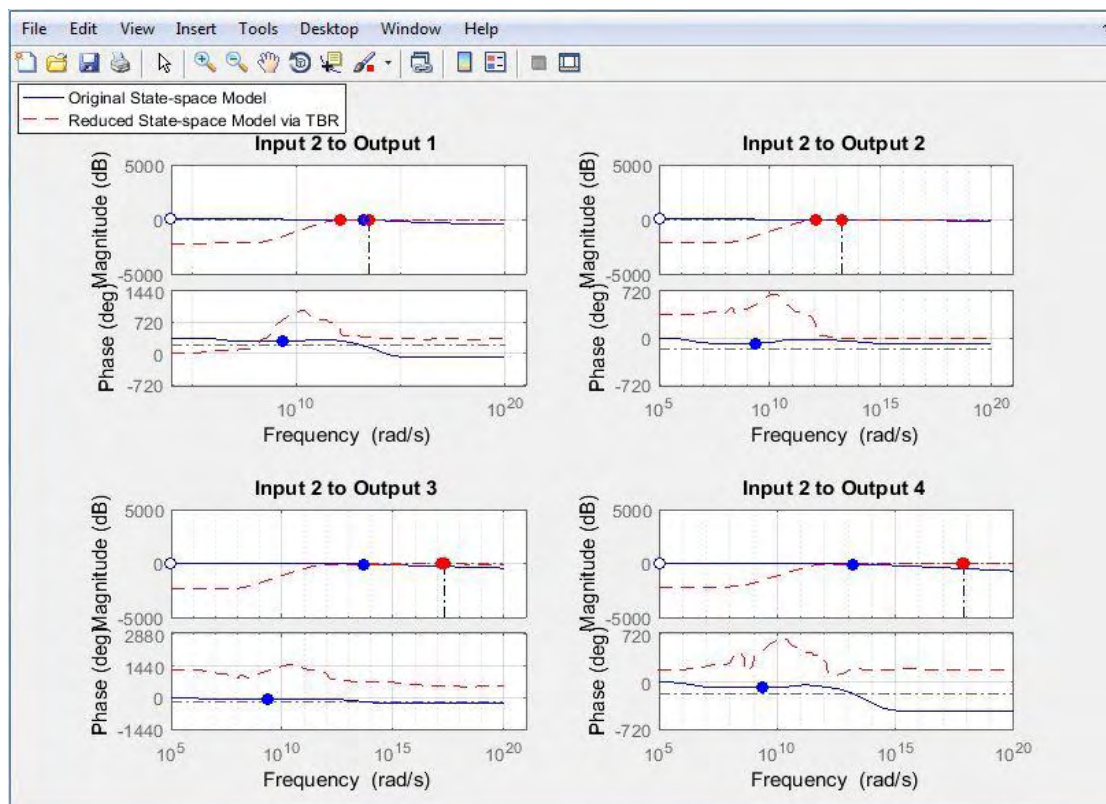
Από τις δύο παραπάνω σχέσεις (7.1, 7.2), παρατηρούμε πως η συνάρτηση μεταφοράς του αρχικού, αλλά και του ελαττωμένου κυκλώματος που προέκυψε από τη μέθοδο υποβιβασμού τάξης μοντέλου, έχουν την ίδια διάσταση.

### Γραφικές παραστάσεις απόκρισης συχνότητας αρχικού – τελικού μοντέλου συστήματος

Στην συνέχεια, παραθέτουμε στην ίδια γραφική παράσταση, ζεύγη ενδεικτικών κυματομορφών της απόκρισης συχνότητας ως προς τη συχνότητα, δηλαδή της συνάρτησης μεταφοράς του αρχικού κυκλώματος και του ελαττωμένου μοντέλου. Συνολικά, παρουσιάζονται 16 γραφικές παραστάσεις για την απόκριση συχνότητας αρχικού και τελικού μοντέλου κυκλώματος, για την καθεμία από τις 4 εισόδους προς καθεμία από τις 4 εξόδους (Εικόνες 33, 34, 35, 36). Επάνω στις κυματομορφές με κόκκινη βούλα έχουμε συμβολίσει το peak response και με μπλε τα minimum stability margins. Παράλληλα, έχουμε οριοθετήσει τις γραφικές παραστάσεις στα πλαίσια grid.



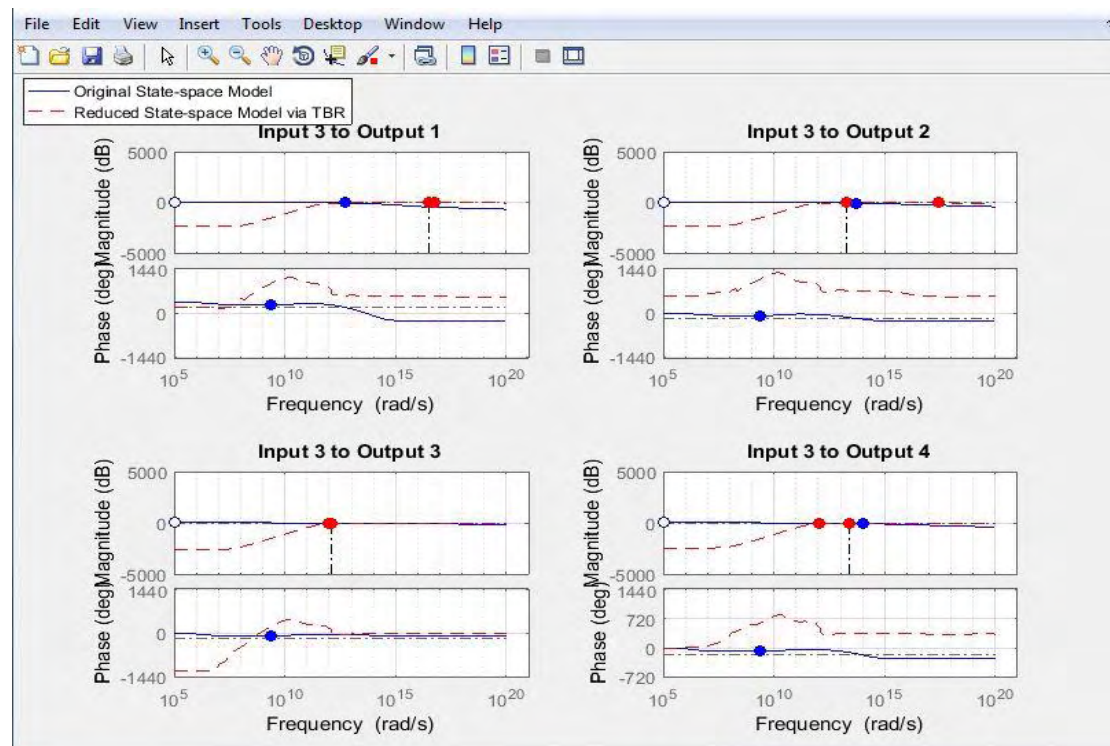
Εικόνα 33: Απόκριση συχνότητας αρχικού - τελικού μοντέλου συστήματος για την 1η είσοδο



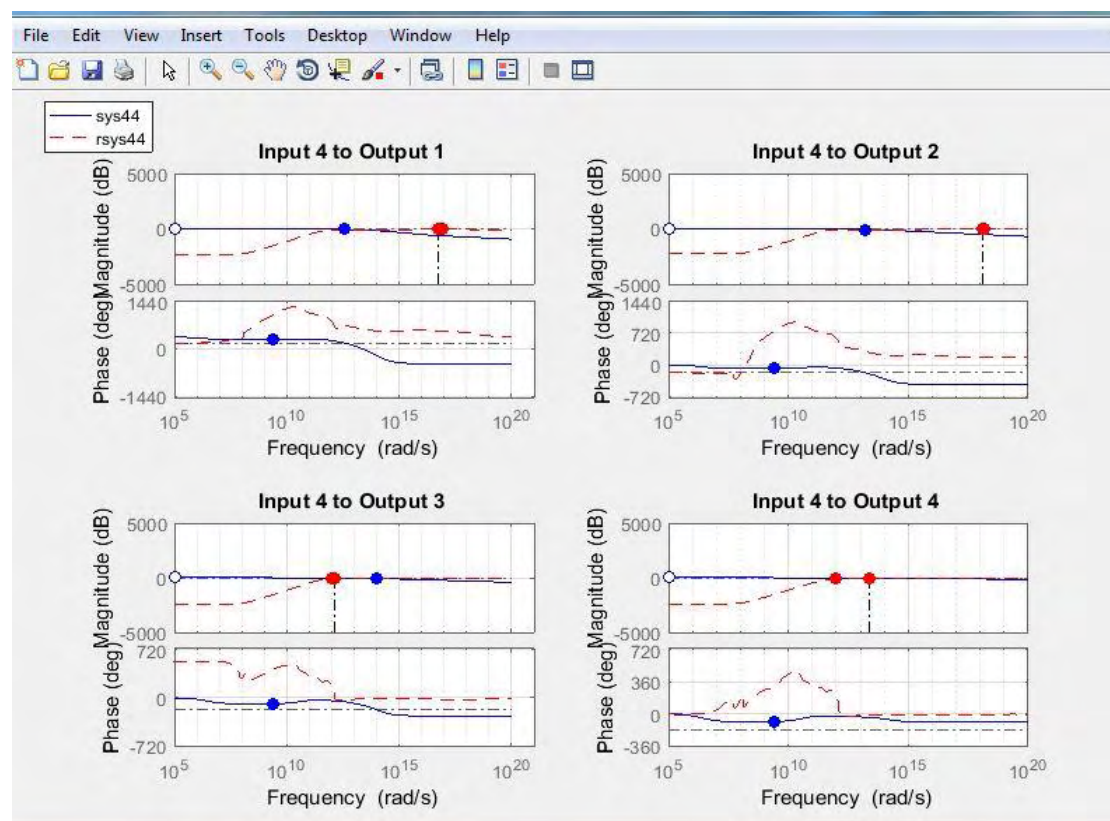
Εικόνα 34: Απόκριση συχνότητας αρχικού - τελικού μοντέλου συστήματος για την 2η είσοδο



## 7.2.2 Ενδεικτικά Αποτελέσματα Προσομοίωσης



Εικόνα 35: Απόκριση συχνότητας αρχικού - τελικού μοντέλου συστήματος για την 3η είσοδο



Εικόνα 36: Απόκριση συχνότητας αρχικού - τελικού μοντέλου συστήματος για την 4η είσοδο

Χρόνοι (σε sec) Εξαγωγής Πινάκων και Προσομοίωσης Κυκλώματος

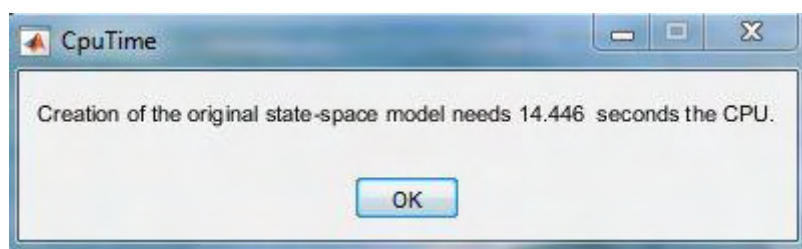
Παρακάτω, φαίνονται οι χρόνοι σε δευτερόλεπτα αναφορικά με την χρήση της CPU που απαιτούνται για να γίνει η εξαγωγή των πινάκων αλλά και οι αντίστοιχοι χρόνοι για την προσομοίωση του κυκλώματος στην κάθε περίπτωση (Πίνακας 7.3). Παρουσιάζονται οι χρόνοι που δαπανώνται για την εξαγωγή πινάκων πριν την ελαχιστοποίηση (original system), μετά από αυτήν (μέσω της minreal) και τέλος κατόπιν της εφαρμογής του MOR αλγόριθμου TBR. Είναι προφανές ότι όσο μικρότερη είναι η τάξη του συστήματος  $q$ , τόσο γρηγορότερα γίνεται η εξαγωγή των πινάκων και η προσομοίωση του συνολικού κυκλώματος. Ο χρόνος εξαγωγής πινάκων έχει μειωθεί σχεδόν κατά το τριπλάσιο μετά από την εφαρμογή του MOR αλγόριθμου. Χρησιμοποιήθηκε η εντολή 'cputime', η οποία επιστρέφει τον συνολικό χρόνο χρήσης CPU σε δευτερόλεπτα από την στιγμή που ξεκίνησε η μέτρηση.

Παρομοίως, με χρήση του ζεύγους εντολών 'tic – toc', λαμβάνουμε τους χρόνους που απαιτούνται για την προσομοίωση του κυκλώματος για  $q = 280$  (original system),  $q = 249$  (ελαχιστοποιήμενο μέσω της minreal),  $q = 224$  (κατόπιν TBR). Το tic σηματοδοτεί την εκκίνηση της μέτρησης και το toc εκτυπώνει τον χρόνο που πέρασε από την στιγμή που έλαβε χώρα η μέτρηση.

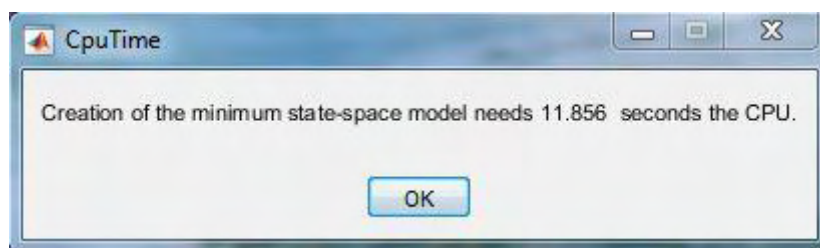
<i>TBR</i>	$q = 280$ (χωρίς ελαχιστοποίηση)	$q = 249$ (με ελαχιστοποίηση)	$q = 224$ (με TBR)
<b>Χρόνος εξαγωγής πινάκων</b>	14.446 sec	11.856 sec	4.914 sec
<b>Χρόνος προσομοίωσης</b>	12.156 sec	6.967 sec	4.740 sec

**Πίνακας 7.3:** Χρόνοι εξαγωγής πινάκων και προσομοίωσης

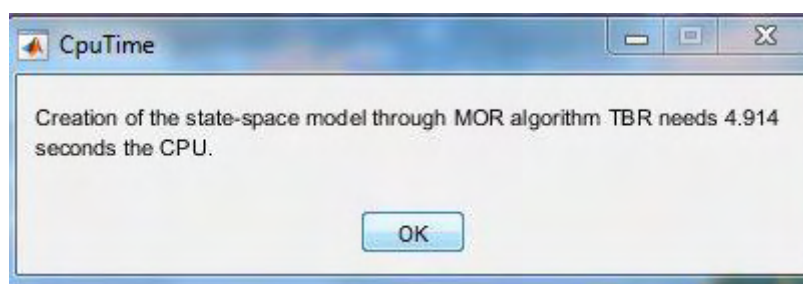
Ακολούθως, φαίνονται τρία παράθυρα διαλόγου (Εικόνες 37, 38, 39) που βλέπει ο χρήστης αναφορικά με τον συνολικό χρόνο που απασχολούμε τον επεξεργαστή στην καθεμία περίπτωση εξαγωγής πινάκων.



**Εικόνα 37:** Χρήση CPU (σε sec) παραγωγής αρχικού μοντέλου συστήματος



Εικόνα 38: Χρήση CPU (σε sec) παραγωγής ελαχιστοποιημένου μοντέλου συστήματος



Εικόνα 39: Χρήση CPU (σε sec) παραγωγής μειωμένου μοντέλου συστήματος μέσω της TBR

### 7.3 Αποτελέσματα μελέτης

Σε αυτό το εδάφιο θα αναλύσουμε τα αποτελέσματα που πήραμε εφαρμόζοντας την πειραματική διαδικασία που περιγράψαμε στην προηγούμενη ενότητα ώστε να αξιολογήσουμε ως σύνολο την μέθοδο TBR. Επιπλέον, θα αναλύσουμε τους χρόνους εξαγωγής πινάκων και προσομοίωσης με σκοπό την αξιολόγηση της υλοποίησής μας.

Μέσα από τα πειράματα που τρέξαμε και τις γραφικές παραστάσεις που πήραμε ως αποτέλεσμα, παρατηρήσαμε πως η συνάρτηση μεταφοράς που αρχικού κυκλώματος και του μετέπειτα ελαττωμένου κυκλώματος συμπίπτουν και προσεγγίζουν ικανοποιητικά η μια την άλλη για καθεμία από τις 4 συνιστώσες του διανύσματος εισόδου του τρέχοντος παραδείγματος. Αυτό βέβαια, μέσα από τη μελέτη που έχουμε κάνει καταλαβαίνουμε πως είναι απόλυτα λογικό και σωστό, επειδή η συμπεριφορά του νέου ελαττωμένου κυκλώματος πρέπει να είναι η ίδια ή αρκετά όμοια με τη συμπεριφορά του αρχικού μας κυκλώματος.

Μέσω του υποβιβασμού τάξης μοντέλου συστήματος καταφέραμε να ελαχιστοποιήσουμε τον χρόνο προσομοίωσης των κυκλωμάτων και να μειώσουμε τόσο την χρήση της CPU (σε δευτερόλεπτα) όσο και της μνήμης που απαιτούταν αρχικά για την προσομοίωση των κυκλωμάτων αυτών. Η χρήση επεξεργαστικής ισχύος (σε sec) μειώνεται ολοένα και περισσότερο όσο μικραίνει η τάξη του συστήματος. Τα αποτελέσματα είναι δεκτά και απολύτως αναμενόμενα, εφόσον έχουμε να επεξεργαστούμε σύστημα μικρότερης κλίμακας και κατά συνέπεια θα δαπανάται λιγότερη επεξεργαστική ισχύς και μνήμη RAM.

Η σελίδα αυτή είναι σκόπιμα λευκή.



# ΚΕΦΑΛΑΙΟ 8

## Επίλογος

Στο τελευταίο κεφάλαιο αυτής της εργασίας, ανακεφαλαιώνουμε τα αποτελέσματα της έρευνας που διεξήγαμε, γίνεται σαφής αναφορά στα πλεονεκτήματα αλλά και τους περιορισμούς της προσέγγισης που ακολουθήθηκε και ταυτόχρονα δίνονται κατευθύνσεις επέκτασής της.

### 8.1 Σύνοψη διπλωματικής εργασίας και των αποτελεσμάτων της

Συνοψίζοντας, στην παρούσα διπλωματική εργασία ασχοληθήκαμε με την χρήση της τεχνικής υποβιβασμού τάξης μοντέλου (MOR), γνωστή ως σύνθεση ισοδύναμου κυκλώματος, ως μέσο για την γρήγορη προσομοίωση των κυκλωμάτων μεγάλης κλίμακας. Οι μέθοδοι υποβιβασμού τάξης μοντέλου που αναλύθηκαν εκτενώς και χρησιμοποιήθηκαν είναι τεχνικές οι οποίες αποσκοπούν στη μείωση της υπολογιστικής πολυπλοκότητας ή της κλίμακας των μαθηματικών μοντέλων κατά τη διάρκεια κάποιας προσομοίωσης.

Αναφερθήκαμε επιγραμματικά στον τρόπο με τον οποίο λειτουργούν διάφοροι αλγόριθμοι της κατηγορίας MOR, επικεντρώνοντας όμως την προσοχή μας στον TBR. Περιγράψαμε αναλυτικά τα βήματα του αλγορίθμου και την διαισθητική του ερμηνεία. Τονίσαμε ότι παρέχει υπολογίσιμα όρια σφάλματος, διατηρεί αναλλοίωτες τις ιδιότητες του αρχικού συστήματος και έχει καθολική ισχύ ανεξαρτήτως της εσωτερικής δομής του πρωταρχικού μοντέλου. Εστιάσαμε κατά κύριο λόγο στο τι είναι μια προσομοίωση, τι χρειάζεται για να γίνει και πως γίνεται να μοντελοποιηθεί ένα κύκλωμα που δίνεται μέσα από ένα αρχείο περιγραφής κυκλώματος. Ωστόσο, ο βασικότερος σκοπός μας κυμάνθηκε στον τρόπο με τον οποίο μπορούν αυτά τα κυκλώματα να λυθούν γρηγορότερα και με τη χρήση όσο το δυνατόν γίνεται λιγότερων πόρων του συστήματος.

Εν κατακλείδι, μέσα από την μελέτη που έγινε πάνω στους αλγορίθμους MOR με την χρήση του TBR καταφέραμε εν τέλει να ελαχιστοποιήσουμε τον χρόνο προσομοίωσης των κυκλωμάτων και να μειώσουμε την τάξη του αρχικού μοντέλου συστήματος. Κατά συνέπεια, περιορίσαμε την χρήση της CPU και της μνήμης που απαιτούνταν αρχικά για την προσομοίωση των κυκλωμάτων αυτών. Ο στόχος μας, ο

οποίος επιτεύχθηκε, ήταν να αποδειχθεί μέσω των διαγραμμάτων Bode ότι η απόκριση συχνότητας του τελικού συστήματος προσεγγίζει όσο το δυνατόν περισσότερο την αντίστοιχη απόκριση του αρχικού μοντέλου με σημαντική εν τούτοις εξοικονόμηση πόρων του υπολογιστικού συστήματος.

## 8.2 Μελλοντικές επεκτάσεις

Μια πιθανή μελλοντική επέκταση της παρούσας εργασίας θα μπορούσε να είναι η εφεύρεση και υλοποίηση ταχύτερων μεθόδων επίλυσης των Lyapunov εξισώσεων, έτσι ώστε να μπορούμε να χειριζόμαστε εξίσου καλά και υψηλής τάξης συστήματα απευθείας, χωρίς να απαιτείται κάποιο προηγούμενο βήμα αναγωγής σε μικρότερης τάξης. Ως γνωστόν, η επίλυση Lyapunov εξισώσεων καθυστερεί κατά πολύ σε μεγάλης κλίμακας συστήματα [10].

Επιπροσθέτως, είναι απαραίτητο και συνάμα ιδιαίτερος χρήσιμο να αναπτυχθεί μια μέθοδος για την επιβολή παθητικότητας και να συμπεριληφθεί στο εργαλείο μας, ώστε να μειωθεί στο ελάχιστο η πιθανότητα κάποιας ασταθούς προσομοίωσης. Υπάρχουν ορισμένες περιπτώσεις όπου ο βασικός αλγόριθμος TBR δεν παράγει παθητικά μοντέλα. Απαιτείται λοιπόν να τον επεκτείνουμε κατάλληλα, ώστε με βεβαιότητα τα καινούρια μοντέλα να είναι παθητικά. Χρειαζόμαστε μια μεθοδολογία με καθολική ισχύ και για αυτό μπορούμε να χρησιμοποιήσουμε το positive-real λήμμα, το οποίο ικανοποιεί τις παρακάτω Lur'e εξισώσεις:

$$AX_c + X_c A^T + K_c K_c^T = 0$$

$$X_c C^T - B + K_c J_c^T = 0$$

$$J_c J_c^T = D + D^T$$

Και τις αντίστοιχες δυϊκές εξισώσεις:

$$A^T X_o + X_o A + K_o^T K_o = 0$$

$$X_o B - C^T + K_o^T J_o = 0$$

$$J_o^T J_o = D + D^T$$

Ως πρώτο βήμα στην θέση των εξισώσεων Lyapunov του TBR επιλύουμε τις παραπάνω εξισώσεις Lur'e για  $X_c$  και  $X_o$  αντίστοιχα. Κατά τα άλλα, ακολουθούμε το ίδιο σκεπτικό με την κλασσική παραδοχή. Κατά συνέπεια, η καινούρια συνάρτηση μεταφοράς βάσει του λήμματος θα είναι positive-real και άρα το σύστημα παθητικό. Μελλοντικά, θα επιδιώξουμε να προσομοιώσουμε ένα κύκλωμα μέσω αυτού του αλγορίθμου (Positive - Real TBR) και να εξετάσουμε τα αποτελέσματά του.

Επιπλέον, ωφέλιμο θα ήταν να επεκταθεί κατάλληλα η βασική εκδοχή του TBR, ώστε εγγυημένα η νέα συνάρτηση μεταφοράς να έχει άνω όριο φράγματος (Bounded - Real TBR). Ως μελλοντικός στόχος τίθεται η προσομοίωση αυτής της παραλλαγής αλγορίθμου και η επαλήθευση των υποθέσεων.

Η παραλληλοποίηση ορισμένων σταδίων της υλοποίησης θα μπορούσε να επιφέρει και αυτή με την σειρά της γρηγορότερα αποτελέσματα στην προσομοίωση.

Στόχος τίθεται επίσης, αφενός η ολοκλήρωση του συστήματος έτσι ώστε να δημιουργήσουμε ένα νέο αρχείο περιγραφής κυκλώματος και αφετέρου η λύση αυτού του νέου κυκλώματος.

Τέλος, η μελέτη της ακρίβειας των δύο λύσεων, του αρχικού και του νέου κυκλώματος, θα μας πείσει για την ισχύ του αλγορίθμου.

Η σελίδα αυτή είναι σκόπιμα λευκή.

# Παραρτήματα

## MATLAB Κώδικες

### A Παράρτημα A.

- [State\\_Space\\_Models.m](#)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Panagiotara E. Paraskevi
%% Bachelor Thesis for Department of Electrical & Computer
%% Engineering of Polytechnic School of University of Thessaly at
%% Volos,Greece.
%% This diploma thesis was conducted during the winter semester of
%% academic year 2016-2017.
%% "Model Order Reduction of Large Scale Circuits(VLSI)with Dynamical
%% System Techniques"
%% Supervisors: Prof.Nestoras Evmorfopoulos, Prof.George Stamoulis

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Starting Point. The simulation of Truncated Balanced
%% Realization(TBR)algorithm was conducted to a small circuit with the
%% follow characteristics in order to be able to come to a conclusion
%% easier:

%% System's Dimension: 280x280
%% Nodes: 281
%% Current Sources: 4 (inputs + outputs)
%% Resistances: 360
%% Capacitors: 280

% % state vector x
% x = [nodes_voltages]';
% nodes_voltages = [1:1:280];

% % uin(t)input vector
% ip(t) = [i1,i2,i3,i4]';

% % uou(t) output vector
% up(t) = [u1,u2,u3,u4]';

clc
clear all
```



```

spy(AA);
A = full(AA);
eigenvaluesA = eig(full(A));
for i = 1:1:size(eigenvaluesA)
    if real(eigenvaluesA(i)) >= 0
        disp('Oops the system is unstable, because matrix A has
            non-negative eigenvalues! Try with a different system.')
        return
    end
end

uiwait(msgbox('Super,the system is stable!','Success','success'))

% Array B is n*Ni(number of state variables and number of inputs
% respectively), is a source connectivity matrix mapping uin(t)
% to the MNA vector x. We put 1 to the positions of matrix where
% we consider ports(inputs/outputs), so that B = L.

BB = zeros(280,4);
BB(3,1) = 1;
BB(5,2)= 1;
BB(42,3)=1;
BB(71,4) = 1;
disp('Array B is: ');
B = full(BB);

% Array R = C^(-1) * BB is nxNi, in MNA format, is the source
% connectivity matrix mapping uin(t) to the MNA vector x.
disp('Array R is: ');
RR = inversedC * BB;
R = full(RR);

% Array L = BB^(T), is the Noxn probing matrix mapping uout(t)
% to the MNA vector x.
disp('Array L is: ');
LL = transpose(BB);
L = full(LL);

D =zeros(4,4);

disp('Origin_A: ')
Orig_A = A;
disp('Origin_B: ')
Orig_B = R;
disp('Origin_C: ')
Orig_C = L;
disp('Origin_D: ')
Orig_D = D;
Orig_Sys = ss(Orig_A, Orig_B, Orig_C, Orig_D)
else
    fprintf('\n\nPlease input the Numerator and Denominator
        Polynomials of your transfer function:\n')
    Orig_Num = input(' Numerator Polynomial: ');
    Orig_Den = input(' Denominator Polynomial: ');
    [Orig_A, Orig_B, Orig_C, Orig_D] = tf2ss(Orig_Num,Orig_Den);
    Orig_Sys = ss(Orig_A, Orig_B, Orig_C, Orig_D);
end

%totalCPUTime1 returns the total CPU time (in seconds) used by
%MATLAB from the time it was started (t1).

```

```

totalCPUTime1 = cputime-t1

fprintf ( 1, 'Creation of the original state-space model needs %1.3f
seconds the CPU.\n', totalCPUTime1 )
msgbox(sprintf('Creation of the original state-space model needs
%1.3f seconds the CPU.\n',totalCPUTime1),'CpuTime')

% end of simulation time for original state-space model
first = toc

%Calculate a minimal realization, removes unobservable and
%uncontrollable states or remove same poles and zeros from denominator
%and numerator polynomials correspondingly.

% start of simulation time for the minimum original state-space model
tic;

t2 = cputime;
syss1 = minreal(Orig_Sys)
%totalCPUTime2 returns the total CPU time (in seconds) used by
%MATLAB from the time it was started (t2).
totalCPUTime2 = cputime-t2

fprintf ( 1, 'Creation of the minimum state-space model needs %1.3f
seconds the CPU.\n', totalCPUTime2 )
msgbox(sprintf('Creation of the minimum state-space model needs %1.3f
seconds the CPU.\n',totalCPUTime2), 'CpuTime')

% end of simulation time for minimum original state-space model
second = toc

fprintf('\n\nWhat is the order of your original MIMO system?\n')
disp('Original Order: ');
Orig_N = rank(Orig_Sys.A)
Orig_minimum = rank(syss1.A)
inputs = rank(Orig_Sys.B)
outputs = rank(Orig_Sys.C)

% check matrix sizes
if size(Orig_Sys.A,2) ~= Orig_N
    error('A must be square.')
end
if size(Orig_Sys.B,1) ~= Orig_N
    error('B must have the same number of rows as A.')
end
if size(Orig_Sys.C,2) ~= Orig_N
    error('C must have the same number of columns as A.')
end

fprintf('\n\nInto which order do you want to reduce your system?\n')
q = -1;
while q<=0 || round(q)~=q
    x = inputdlg('Enter desired reduction order:',...
        'Reduction Order', [1 50]);
    q = str2num(x{:});
    if q<=0 || round(q)~=q
        h1 = msgbox('Invalid Value', 'Error','error');
        msgbox('You must input as Reduction Error a positive value.',
            'Reminder' );
    end
end

```



```

        continue
    end
    break
end

fprintf('\n\nWhich algorithm would you like to use to perform the
reduction? (type 1 or 2)\n')
fprintf('\t1.Standard Approach of Truncated Balanced Realization
(TBR).\n')
fprintf('\t2.DC - Accuracy TBR. \n')
meth = input('Answer: ');

if meth == 1
    method = 'tbr'
else
    method = 'matchDcGain'
end

%Create a structure and define as a field the upper error bound
%(statically). After trials, we end up that the best approximation
%between the initial and the reduced model is received when redError
%is equal to 10e-15, so redErr = 10e-15.

s = struct('type', method, 'redErr',10e-
15, 'hsvTol',0, 'warnorError', 'warn')

fprintf('\n\nWhich of the following function calls would you like to
use? -the difference relies on input argumnets- (type 1, 2, 3 or
4)\n')
fprintf('\t1. Input argument: System.\n')
fprintf('\t2. Input argument: System and system order.\n')
fprintf('\t3. Input argument: System and struct.\n')
fprintf('\t4. Input argument: System, system order and struct.\n')
value = input('Answer: ');

switch s.type

%%%%%%%%%%%%%%
% TBR Algorithm %
%%%%%%%%%%%%%%
    case 'tbr'
        sys1 = syss1;
%%%%%%%%%%%%%%
% matchDcGain Algorithm %
%%%%%%%%%%%%%%
    case 'matchDcGain'
        Anew = inv(syss1.A)
        Bnew = -Anew*syss1.B
        Cnew = syss1.C*Anew
        Dnew = syss1.D - syss1.C*Anew*syss1.B
        sys1 = ss(Anew,Bnew,Cnew,Dnew)
end

if value == 1
%read the new system order from keyboard in function TBR,
%maybe not very good approximation to the initial model, as no
%reduction upper bound was defined.
    [rsys1,q,TBalInverse,TBal,HSV] = tbr_final(sys1)

```

```

elseif value == 2
%read the new system order from keyboard above, maybe not very good
%approximation to the initial model, as no reduction upper bound was
%defined.
[rsys1,q,TBalInverse,TBal,HSV] = tbr_final(sys1,q)
elseif value == 3
%the new system's order is computed according to the reduction error,
%which in our case is defined statically: redErr = 10e-15. This redErr
%is proved that gives the best approximation to the initial model
%amongst all.
[rsys1,q,TBalInverse,TBal,HSV] = tbr_final(sys1,s)
else
%read the new system order from keyboard above, maybe not very good
%approximation to the initial model, as the given redError=10e-15
%isn't took into account (being ignored).
[rsys1,q,TBalInverse,TBal,HSV] = tbr_final(sys1,q,s)
end

%% Return from Model Order Reduction Algorithm TBR
uiwait(msgbox('Procedure of Model Order Reduction is completed
successfully! Lets design the Bode plot!','Success','modal'))

%% Plotting the frequency response (Bode plot) of the original and
reduced models.

%% MIMO system, plot for the 1st input to each one of the vector's
outputs - size(outputs).

%% 1st input
sys11 = sys1(1,1);
sys12 = sys1(1,2);
sys13 = sys1(1,3);
sys14 = sys1(1,4);

rsys11 = rsys1(1,1);
rsys12 = rsys1(1,2);
rsys13 = rsys1(1,3);
rsys14 = rsys1(1,4);

legend('Original State-space Model', 'Reduced State-space Model via
TBR')

% % Bode plot of the frequency response
figure(1)
subplot(2,2,1);
bodeplot(sys11,'b',rsys11,'r--')
title('Input 1 to Output 1')

subplot(2,2,2);
bodeplot(sys12,'b',rsys12,'r--')
title('Input 1 to Output 2')

subplot(2,2,3);
bodeplot(sys13,'b',rsys13,'r--')
title('Input 1 to Output 3')

subplot(2,2,4);
bodeplot(sys14,'b',rsys14,'r--')
title('Input 1 to Output 4')

```

```

%% 2nd input
sys21 = sys1(2,1);
sys22 = sys1(2,2);
sys23 = sys1(2,3);
sys24 = sys1(2,4);

rsys21 = rsys1(2,1);
rsys22 = rsys1(2,2);
rsys23 = rsys1(2,3);
rsys24 = rsys1(2,4);

legend('Original State-space Model', 'Reduced State-space Model via
TBR')

% % Bode plot of the frequency response
figure(2)
subplot(2,2,1);
bodeplot(sys21,'b',rsys21,'r--')
title('Input 2 to Output 1')

subplot(2,2,2);
bodeplot(sys22,'b',rsys22,'r--')
title('Input 2 to Output 2')

subplot(2,2,3);
bodeplot(sys23,'b',rsys23,'r--')
title('Input 2 to Output 3')

subplot(2,2,4);
bodeplot(sys24,'b',rsys24,'r--')
title('Input 2 to Output 4')

%% 3rd input
sys31 = sys1(3,1);
sys32 = sys1(3,2);
sys33 = sys1(3,3);
sys34 = sys1(3,4);

rsys31 = rsys1(3,1);
rsys32 = rsys1(3,2);
rsys33 = rsys1(3,3);
rsys34 = rsys1(3,4);

legend('Original State-space Model', 'Reduced State-space Model via
TBR')

% % Bode plot of the frequency response
figure(3)
subplot(2,2,1);
bodeplot(sys31,'b',rsys31,'r--')
title('Input 3 to Output 1')

subplot(2,2,2);
bodeplot(sys32,'b',rsys32,'r--')
title('Input 3 to Output 2')

subplot(2,2,3);
bodeplot(sys33,'b',rsys33,'r--')
title('Input 3 to Output 3')

```

```

subplot(2,2,4);
bodeplot(sys34,'b',rsys34,'r--')
title('Input 3 to Output 4')

%% 4th input
sys41 = sys1(4,1);
sys42 = sys1(4,2);
sys43 = sys1(4,3);
sys44 = sys1(4,4);

rsys41 = rsys1(4,1);
rsys42 = rsys1(4,2);
rsys43 = rsys1(4,3);
rsys44 = rsys1(4,4);

legend('Original State-space Model', 'Reduced State-space Model via
TBR')

% % Bode plot of the frequency response
figure(4)
subplot(2,2,1);
bodeplot(sys41,'b',rsys41,'r--')
title('Input 4 to Output 1')

subplot(2,2,2);
bodeplot(sys42,'b',rsys42,'r--')
title('Input 4 to Output 2')

subplot(2,2,3);
bodeplot(sys43,'b',rsys43,'r--')
title('Input 4 to Output 3')

subplot(2,2,4);
bodeplot(sys44,'b',rsys44,'r--')
title('Input 4 to Output 4')

%% Compute infinite norm of the difference of the two transfer
functions.
E = sys1 - rsys1;
H_Norm = norm(E);
Hinf_Norm = norm(E,inf)
fprintf('The Hinf-Norm of the Reduced Model: %.9f\n\n',Hinf_Norm)

fprintf('\n\nDo you want to reduce another model? (type 1 or 2)\n')
fprintf('\t1. Yes I do.\n')
fprintf('\t2. No I dont. Exit program.\n')
X = input('Answer: ');

if X == 1
    uiwait(msgbox('Please run again this function and enter the new
system model for reduction.', 'Reminder','reminder'))
else
    uiwait(msgbox('Thank you for using our Model Order Reduction
algorithm. For your comments and suggestions please email me at:
parpanag@inf.uth.gr. Copyright © Panagiotara E. Paraskevi, 2017.
All rights reserved (R).', 'End of Procedure','end of
procedure'))
end

```

- [tbr\\_final.m](#)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%% Panagiotara E. Paraskevi
%% Bachelor Thesis for Department of Electrical & Computer Engineering
%% of Polytechnic School of University of Thessaly at Volos,Greece.
%% This diploma thesis was conducted during the winter semester of
%% academic year 2016-2017.
%% "Model Order Reduction of Large Scale Circuits(VLSI)with Dynamical
%% System Techniques"
%% Supervisors: Prof.Nestoras Evmorfopoulos, Prof.George Stamoulis
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%Start
```

```
function [rsys1, varargout] = tbr_final(sys1, varargin)
```

```
% Description:
```

```
%     Computes a reduced model of order q by balancing and truncation,
%     i.e. by transforming the system to a balanced realization where
%     all states are equally controllable and observable and selecting
%     only the first q modes responsible for the highest energy
%transfer in system.
```

```
%
%     If q is specified as system order, then TBR computes a balanced
% realization of the system.
```

```
%
%     If a reduction order q is passed to the function, the reduced
% system will be of this size with the options 'hsvTol' and
% 'redErr' ignored. If not, the option reduction error 'redErr'
% is crucial. This error is defined as two times the sum of all
% Hankel-Singular values bigger than the reduction order. To
% avoid this option it can be set to zero ('redErr'=0). If so,
% the Hankel-Singular values (satisfying the option 'hsvTol')
% will be plotted for the user to enter a desired reduction
% order. If the option 'type' is set to 'matchDcGain', then a
% residualization is computed to match the DC gain of the
% original model.
```

```
%
% Input Arguments:
%     Required Input Arguments:
%     -sys:   an ss - object containing the LTI system
%     Optional Input Arguments:
%     -q:     order of reduced system
%     -Opts:  a structure containing following options
%             -.type:      select amongst different algorithms
%                         'tbr' / 'matchDcGain'
%             -.redErr:    upper bound of reduction error
%             -.hsvTol:    tolerance for Hankel-Singular values
%             -.warnOnError: display warnings or errors
```

```
%
% Output Arguments:
%     -sysr:   reduced system
%     -q:      order of reduced system
%     -TBalInverse,TBal:  projection matrices
```

```

%      -hsv:   Hankel singular values

% Default execution parameters
Default.type = 'tbr'; % default method:'tbr'
Default.redErr = 0; % redErr>2*sum(hsv(q+1:end)))
Default.hsvTol = 1e-15; % hsv tolerance (hsv(q)<hsvTol)
Default.warnOnError = 'warn'; % display warnings or errors
(0,'warn','error')

% check input arguments for order q and Opts
if nargin>1
    if nargin == 2 && ~isa(varargin{1},'double')
        Opts = varargin{1};
    else
        q = varargin{1};
        if nargin == 3
            Opts = varargin{2};
        end
    end
end

% create the options structure
if ~exist('Opts','var') || isempty(Opts)
    Opts = Default
else
    if ~isfield(Opts,'type')
        Opts.type = 'tbr'
    end
    Opts
end

% main body of Truncated Balanced Realization procedure
if strcmp(Opts.type,'tbr') || strcmp(Opts.type,'matchDcGain')
    qmax1 = rank(sys1.A)
    lyapOpts.type = 'builtIn';
    if ~exist('R','var') || ~exist('L','var')

% First computes the sollution Wc of the Lyaponov equation
%  $A*W_c + W_c*A' = -B*B'$  and then analyzes Wc into Cholesky factors
% so that  $W_c = L_c*L_c'$ ---where A and Bnewest represent square matrices
% of identical sizes. If Bnewest is a symmetric matrix,
% the solution Wc is also a symmetric matrix.

        Bnewest = full(sys1.B * sys1.B');
        Wc = lyap(sys1.A,Bnewest);
        Wceigenvalues = eig(Wc);
        LLc = nearestSPD(Wc);
        Lc = chol(LLc);
        R = Lc;

%Then computes the solution Wo of the Lyaponov equation
%  $A'*W_o + W_o*A = -C'*C$  and after that analyzes Wo into Cholesky
%factors so that  $W_o = L_o*L_o'$ ---where A and Cnewest represent square
%matrices of identical sizes. If Cnewest is a symmetric matrix, the
%solution Wo is also a symmetric matrix.

        Cnewest = full(sys1.C' * sys1.C);
        Wo = lyap(sys1.A,Cnewest);
        Woeigenvalues = eig(Wo);
        LLo = nearestSPD(Wo);
    end
end

```

```

        Lo = chol(LLo);
        L = Lo;
    end
end

% calculate balancing transformation and Hankel Singular Values
[U,S,V] = svd(full(R*L'));
hsv = diag(S);
HankelSingularValues = real(hsv);
TBalInv = diag(ones(size(hsv))./sqrt(hsv))*U'*L';
TBal = R*V*diag(ones(size(hsv))./sqrt(hsv));

% determine reduction order
if exist('q','var') || Opts.redErr>0
    if ~exist('q','var')
        if strcmp(Opts.type,'tbr') || strcmp(Opts.type,'matchDcGain')
            hsvSum = 0
        end
        for i = qmax1:-1:0
            if hsvSum>Opts.redErr || i==0
                q = i+1
                if q>qmax1
                    q = qmax1
                end
                break;
            else
                hsvSum=hsvSum+2*real(hsv(i))/real(hsv(1))
            end
        end

        if strcmp(Opts.type,'tbr') || strcmp(Opts.type,'matchDcGain')
            warning(['Reduction order was set to q = ',
                num2str(q,'%d'),...' in order to satisfy the upper bound
                for the reduction error. ']);
        end

        if q>rank(sys1.A)
            if strcmp(Opts.warnOnError,'error')
                h1 = msgbox('Reduction order exceeds system order.')
                error('Reduction order exceeds system order.')
            elseif strcmp(Opts.warnOnError,'warn')
                warning(['Reduction order exceeds system order. q has
                been changed to ',...'the system order qmax = ',
                num2str(qmax,'%d'), '.']);
            end
            q=rank(sys1.A);
        end
    end
else
    % The reduction order should be read from terminal,
    % because no order is given as an input argument and redErr < 0
    % or redErr = 0,too.
    qmax = min([sum(hsv>=Opts.hsvTol*hsv(1)), qmax1])
    h = figure(1)
    bar(1:qmax,abs(hsv(1:qmax))./hsv(1),'r')
    title('Hankel Singular Values');
    xlabel('Order');
    ylabel(['Relative hsv decay';sprintf('abs(hsv/hsv(1)) with
    hsv(1)=%.4d',hsv(1))]);
end

```

```

set(gca,'YScale','log');
set(gca, 'YLim', [-Inf;1.5]);
set(gca, 'XLim', [0; qmax]);

q = -1;
while q<0 || round(q)~=q
    prompt = ['Please enter the desired order: (0<= q <= ',
num2str(qmax,'%d'),' '];
    q = input(prompt)

    if q>rank(sys1.A) && qmax==rank(sys1.A)
        if strcmp(Opts.warnOnError,'error')
            error('Reduction order exceeds system order.')
        elseif strcmp(Opts.warnOnError,'warn')
            warning(['Reduction order exceeds system order. q has
been changed to ',... 'the system order qmax = ',
num2str(qmax,'%d'), '.'])
        end
        q = qmax
    elseif q>qmax
        if strcmp(Opts.type,'tbr') || strcmp(Opts.type,'matchDcGain')
            if strcmp(Opts.warnOnError,'error')
                error(['Reduction order must be smaller than q =
', num2str(qmax,'%d'),... ' due to Hankel-Singular values smaller than
the given tolerance',... ' (see Opts.hsvTol).'])
            elseif strcmp(Opts.warnOnError,'warn')
                warning(['q has been changed to qmax = ',
num2str(qmax,'%d'),... ' due to Hankel-Singular values smaller than
the given '...'tolerance (see Opts.hsvTol).'])
            end
        end
        q = qmax
    end

    if q<0 || round(q)~=q
        h = msgbox('Invalid Value.Try again!', 'Error','error')
        continue
    end
end

if ishandle(h)
    close Figure 1;
end

end

%% compute Balanced Realization of matrices A,B,C,D

disp('Resulting balanced reduced order system: ')
ABal = TBalInv*sys1.A*TBal;
BBal = TBalInv*sys1.B;
CBal = sys1.C*TBal;
DBal = sys1.D;

%%partition matrices A,B,C,D conformally, according to the new
%model's order q

A11 = ABal(1:q,1:q);
A12 = ABal(1:q,q+1:end);

```



```

A21 = ABal(q+1:end,1:q);
A22 = ABal(q+1:end,q+1:end);

B1 = BBal(1:q,:);
B2 = BBal(q+1:end,:);
C1 = CBal(:,1:q);
C2 = CBal(:,q+1:end);

D = zeros(4,4);

%%Truncate the transformed matrices A,B,C,D, so as to retain only
%those that represent the strong sub-systems and delete the weak sub-
%systems. The strong sub-systems can contribute the biggest amount of
%energy to the system.

switch Opts.type
    case 'tbr'
        A_Red = A11;
        B_Red = B1;
        C_Red = C1;
        D_Red = D;
        % form the reduced realization to be returned
        fprintf('\n\nThe State Space Representation of the resulting
Reduced Model:\n')
    case 'matchDcGain'
        disp('Perform the inverse reciprocal transformation')
        A_Red = inv(A11);
        B_Red = -A_Red*B1;
        C_Red = C1*A_Red;
        D_Red = D - C1*A_Red*B1;
        % form the reduced reciprocal realization to be returned
        fprintf('\n\nThe State Space Reciprocal Representation of the
resulting Reduced Model:\n')
end

% start of simulation time for reduced (via TBR) state-space model
tic;

t3 = cputime;
rsys1 = ss(A_Red,B_Red,C_Red,D_Red)
% totalCPUTime3 returns the total CPU time (in seconds) used by
MATLAB from the time it was started (t3).
totalCPUTime3 = cputime-t3

fprintf ( 1, 'Creation of the state-space model through MOR algorithm
TBR needs %.4f seconds the CPU.\n', totalCPUTime3 );
msgbox(sprintf('Creation of the state-space model through MOR
algorithm TBR needs %1.3f seconds the
CPU.\n',totalCPUTime3),'CpuTime')

% end of simulation time for reduced (via TBR) state-space model
third = toc

% Reduced size matrices of the returned state-space model system
AredSize = size(rsys1.A)
BredSize = size(rsys1.B)
CredSize = size(rsys1.C)
DredSize = size(rsys1.D)

```

```

eigenvalues = eig(A_Red);

% Output arguments
if nargout>1
    varargout{1} = q;
    varargout{2} = TBalInv;
    varargout{3} = TBal;
    if nargout>3
        varargout{4} = HankelSingularValues;
        varargout{5} = R;
        varargout{6} = L;
    end
end
end

```

- [MatrixMarketRead.m](#)

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Panagiotara E. Paraskevi
%% Bachelor Thesis for Department of Electrical & Computer
%% Engineering of Polytechnic School of University of Thessaly
%% at Volos,Greece.
%% This diploma thesis was conducted during the winter semester of
%% academic year 2016-2017.
%% "Model Order Reduction of Large Scale Circuits(VLSI)with Dynamical
%% System Techniques"
%% Supervisors: Prof.Nestoras Evmorfopoulos, Prof.George Stamoulis

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [A,rows,cols,entries,rep,field,symm] = MatrixMarketRead(filename)
%
% function [A] = mmread(filename)
%
% function [A,rows,cols,entries,rep,field,symm] = mmread(filename)
%
% Reads the contents of the Matrix Market file 'filename'
% into the matrix 'A'. 'A' will be either sparse or full,
% depending on the Matrix Market format indicated by
% 'coordinate' (coordinate sparse storage), or
% 'array' (dense array storage). Branches
% suitably, if real, complex or pattern entries are read in the
% input file. The data will be duplicated as appropriate if
%symmetry is indicated in the header (general, symmetric, Hermitian)
%
% Optionally, size information about the matrix can be
% obtained by using the return values rows, cols, and
% entries, where entries is the number of nonzero entries
% in the final matrix. Type information can also be retrieved
% using the optional return values rep (representation), field,
% and symm (symmetry).
%

mmfile = fopen(filename,'r');
if ( mmfile == -1 )

```

```

    disp(filename);
    error('File not found');
end;

header = fgets(mmfile);
if (header == -1 )
    error('Empty file.')
end

[head0,header] = strtok(header);
[head1,header] = strtok(header);
[rep,header] = strtok(header);
[field,header] = strtok(header);
[symm,header] = strtok(header);
head1 = lower(head1);
rep = lower(rep);
field = lower(field);
symm = lower(symm);
if ( length(symm) == 0 )
    disp(['Not enough words in header line of file ',filename])
    disp('Recognized format: ')
    disp('%%MatrixMarket matrix representation field symmetry')
    error('Check header line.')
end
if ( ~ strcmp(head0,'%%MatrixMarket') )
    error('Not a valid MatrixMarket header.')
end
if ( ~ strcmp(head1,'matrix') )
    disp(['This seems to be a MatrixMarket ',head1,' file.']);
    disp('This function only knows how to read MatrixMarket matrix
files.');
```

```

    disp(' ');
    error(' ');
end

% Read through comments, ignoring them

commentline = fgets(mmfile);
while length(commentline) > 0 & commentline(1) == '%',
    commentline = fgets(mmfile);
end

% Read size information, then branch according to
% sparse or dense format

if ( strcmp(rep,'coordinate')) % read matrix given in sparse
                                % coordinate matrix format

    [sizeinfo,count] = sscanf(commentline,'%d%d%d');
    while ( count == 0 )
        commentline = fgets(mmfile);
        if (commentline == -1 )
            error('End-of-file reached before size information was
found.')
```

```

        end
        [sizeinfo,count] = sscanf(commentline,'%d%d%d');
        if ( count > 0 & count ~= 3 )
            error('Invalid size specification line.')
        end
    end
end
end

```

```

rows = sizeinfo(1);
cols = sizeinfo(2);
entries = sizeinfo(3);

if ( strcmp(field,'real') ) % real valued entries:

    [T,count] = fscanf(mmfile,'%f',3);
    T = [T; fscanf(mmfile,'%f')];
    if ( size(T) ~= 3*entries )
        message = ...
        str2mat('Data file does not contain expected amount of
data.',... 'Check that number of data lines matches nonzero count. ');
        disp(message);
        error('Invalid data. ');
    end

    T = reshape(T,3,entries)';
    disp('Array is: ');
    A = (sparse(T(:,2)+1, T(:,1)+1, T(:,3), rows , cols))';
    isequal(A,A. ');

elseif ( strcmp(field,'complex') ) % complex valued entries:

    T = fscanf(mmfile,'%f',4);
    T = [T; fscanf(mmfile,'%f')];
    if ( size(T) ~= 4*entries )
        message = ...
        str2mat('Data file does not contain expected amount of
data.',... 'Check that number of data lines matches nonzero count. ');
        disp(message);
        error('Invalid data. ');
    end

    T = reshape(T,4,entries)';
    A = sparse(T(:,1), T(:,2), T(:,3) + T(:,4)*sqrt(-1), rows ,
cols);

elseif ( strcmp(field,'pattern') ) % pattern matrix (no values given):

    T = fscanf(mmfile,'%f',2);
    T = [T; fscanf(mmfile,'%f')];
    if ( size(T) ~= 2*entries )
        message = ...
        str2mat('Data file does not contain expected amount of
data.',... 'Check that number of data lines matches nonzero count. ');
        disp(message);
        error('Invalid data. ');
    end

    T = reshape(T,2,entries)';
    A = sparse(T(:,1), T(:,2), ones(entries,1) , rows , cols);

end

elseif ( strcmp(rep,'array') ) % read matrix given in dense
% array (column major) format

[sizeinfo,count] = sscanf(commentline,'%d%d');
while ( count == 0 )
    commentline = fgets(mmfile);
    if (commentline == -1 )

```

```

        error('End-of-file reached before size information was
found.')
```

```

    end
    [sizeinfo,count] = sscanf(commentline,'%d%d');
    if ( count > 0 & count ~= 2 )
        error('Invalid size specification line.')
```

```

    end
end
rows = sizeinfo(1);
cols = sizeinfo(2);
entries = rows*cols;
if ( strcmp(field,'real') )                                % real valued entries:
    A = fscanf(mmfile,'%f',1);
    A = [A; fscanf(mmfile,'%f')];
    if ( strcmp(symm,'symmetric') | strcmp(symm,'hermittian') |
strcmp(symm,'skew-symmetric') )
        for j=1:cols-1,
            currenti = j*rows;
            A = [A(1:currenti); zeros(j,1);A(currenti+1:length(A))];
        end
    elseif ( ~ strcmp(symm,'general') )
        disp('Unrecognized symmetry')
        disp(symm)
        disp('Recognized choices:')
        disp('    symmetric')
        disp('    hermitian')
        disp('    skew-symmetric')
        disp('    general')
        error('Check symmetry specification in header.');
```

```

    end
    A = reshape(A,rows,cols);
elseif ( strcmp(field,'complex') )                        % complex valued entries:
    tmp = fscanf(mmfile,'%f',1);
    tmpi = fscanf(mmfile,'%f',1);
    A = tmp+tmpi*i;
    for j=1:entries-1
        tmp = fscanf(mmfile,'%f',1);
        tmpi = fscanf(mmfile,'%f',1);
        A = [A; tmp + tmpi*i];
    end
    if ( strcmp(symm,'symmetric') | strcmp(symm,'hermitian') |
strcmp(symm,'skew-symmetric') )
        for j=1:cols-1,
            currenti = j*rows;
            A = [A(1:currenti); zeros(j,1);A(currenti+1:length(A))];
        end
    elseif ( ~ strcmp(symm,'general') )
        disp('Unrecognized symmetry')
        disp(symm)
        disp('Recognized choices:')
        disp('    symmetric')
        disp('    hermitian')
        disp('    skew-symmetric')
        disp('    general')
        error('Check symmetry specification in header.');
```

```

    end
    A = reshape(A,rows,cols);
elseif ( strcmp(field,'pattern') )                        % pattern (makes no sense for
dense)
    disp('Matrix type:',field)
    error('Pattern matrix type invalid for array storage format.');
```

```

else                                     % Unknown matrix type
    disp('Matrix type:',field)
    error('Invalid matrix type specification. Check header against MM
documentation.');
```

end

end

```

% If symmetric, skew-symmetric or Hermitian, duplicate lower
% triangular part and modify entries as appropriate

if ( strcmp(symm,'symmetric') )
    A = A + A.' - diag(diag(A));
    entries = nnz(A);
elseif ( strcmp(symm,'hermitian') )
    A = A + A' - diag(diag(A));
    entries = nnz(A);
elseif ( strcmp(symm,'skew-symmetric') )
    A = A - A';
    entries = nnz(A);
end

fclose(mmfile);
```

- [nearestSPD.m](#)

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Panagiotara E. Paraskevi
%% Bachelor Thesis for Department of Electrical & Computer
%% Engineering of Polytechnic School of University of Thessaly
%% at Volos,Greece.
%% This diploma thesis was conducted during the winter semester of
%% academic year 2016-2017.
%% "Model Order Reduction of Large Scale Circuits(VLSI) with
%% Dynamical System Techniques"
%% Supervisors: Prof.Nestoras Evmorfopoulos, Prof.George Stamoulis

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Ahat = nearestSPD(A)

% nearestSPD - the nearest Symmetric Positive
% Definite matrix to A usage: Ahat = nearestSPD(A)
%
% From Higham: "The nearest symmetric positive semidefinite matrix in
% the Frobenius norm to an arbitrary real matrix A is shown to be (B
% + H)/2, where H is the symmetric polar factor of B=(A + A')/2."
%
% Input Arguments
%   A square matrix, which will be converted to the nearest Symmetric
%   Positive Definite Matrix.
%
% Output Arguments
```

```

% Ahat is the matrix chosen as the nearest SPD matrix to A.

%% %%

if nargin ~= 1
    error('Exactly one argument must be provided.')
end

% test for a square matrix A
[r,c] = size(A);
if r ~= c
    error('A must be a square matrix.')
elseif (r == 1) && (A <= 0)
    % A was scalar and non-positive, so just return eps
    Ahat = eps;
    return
end

% symmetrize A into B
B = (A + A')/2;

% Compute the symmetric polar factor of B. Call it H.
% Clearly H is itself SPD.
[U,Sigma,V] = svd(B);
H = V*Sigma*V';

% get Ahat in the above formula
Ahat = (B+H)/2;

% ensure symmetry
Ahat = (Ahat + Ahat')/2;

% test that Ahat is in fact PD. if it is not so, then tweak it just a
bit.
p = 1;
k = 0;
% if p = 0 then Ahat is SPD, else p != 0 Ahat is not SPD.
while p ~= 0
    [R,p] = chol(Ahat);
    k = k + 1;
    if p ~= 0
        % Ahat failed the chol test. It must have been just a hair off,
        % due to floating point trash, so it is simplest now just to
        % tweak by adding a tiny multiple of an identity matrix.
        mineig = min(eig(Ahat));
        Ahat = Ahat + (-mineig*k.^2 + eps(mineig))*eye(size(A));
    end
end
end

```

# Βιβλιογραφία

- [1] “*Guaranteed Passive Balancing Transformations for Model Order Reduction*”, Joel R. Phillips, Member IEEE, Luca Daniel, Student Member IEEE and L. Miguel Silveira, Senior Member IEEE.
- [2] “*IC interconnect analysis*”, Celik, Mustafa, Lawrence Pileggi and Altan Odabasioglu, Springer Science and Business Media, 2002.
- [3] “*Εισαγωγή στη θεωρία σημάτων και συστημάτων*”, Θεοδωρίδης, Σ., Κ. Μπερμπερίδης, και Α. Κοφίδης. Αθήνα: Τυπωθήτω-Εκδόσεις Γιώργου Δαρδανού (2003).
- [4] “*Σημειώσεις μαθήματος “Προσομοίωση Κυκλωμάτων”*”, Διδάσκων Ευμορφόπουλος Νέστωρ, Φθινόπωρο 2011, Τμήμα Μηχανικών Η/Υ Τηλεπικοινωνιών και Δικτύων, Πανεπιστήμιο Θεσσαλίας.
- [5] “*A Sequential Reduction Scheme in Reduced Order Modelling of Large Scale Systems*”, Behnam Salimbahrami, Institute of Automatic Control, Technische Universitat Munchen, Germany.
- [6] “*Balanced realization of lossless systems: Schur parameters, canonical forms and Applications*”, Ralf Peeters, Martine Olivi, Bernard Hanzon.
- [7] “*A survey of model reduction by balanced truncation and some new results*”, Serkan Gugercin, Athanasios C. Antoulas.
- [8] “*Introduction to Model Order Reduction*”, Will Schilders, Eindhoven, The Netherlands, Eindhoven University of Technology, Faculty of Mathematics and Computer Science.
- [9] “*A survey of model reduction methods for large-scale systems*”, A.C. Antoulas, D.C. Sorensen and S. Gugercin.
- [10] “*An Efficient Lyapunov Equation-Based Approach for Generating Reduced-Order Models of Interconnect*”, Jing-Rebecca Li, Frank Wang, Jacob White, Research Laboratory of Electronics, Massachusetts Institute of Technology.
- [11] “*Balanced Truncation Model Reduction for Large-Scale Systems in Descriptor Form*”, Volker Mehrmann, Tatjana Stykel.
- [12] [https://en.wikipedia.org/wiki/Symmetric\\_matrix](https://en.wikipedia.org/wiki/Symmetric_matrix)
- [13] [https://en.wikipedia.org/wiki/Positive-definite\\_matrix](https://en.wikipedia.org/wiki/Positive-definite_matrix)



- [14] [https://en.wikipedia.org/wiki/Modified\\_nodal\\_analysis](https://en.wikipedia.org/wiki/Modified_nodal_analysis)
- [15] <http://math.nist.gov/MatrixMarket/formats.html>
- [16] “*Sequential Reduction Scheme in Reduced Order Modelling of Large Scale Systems*”, Behnam Salimbahrami, Institute of Automatic Control, Technische Universität München.
- [17] “*Order Reduction of Linear Dynamic System using Matlab Programming Method*”, D.Devi and P.Poongodi, Electronics and Instrumentation, Karunya University Coimbatore.
- [18] “*Model reduction of modular systems using balancing methods*”, Andreas Keil, Jean-Luc Gouzé.
- [19] “*An efficient DC-gain matched balanced truncation realization for VLSI interconnect circuit order reduction*”, Xuan Zenga, Dian Zhou, Wei Caic, Microelectronic Department, Fudan University, Shanghai 200433, PR China.
- [20] “*Balanced Truncation Model Order Reduction for LTI Systems with many Inputs or Outputs*”, Peter Benner and André Schneider, Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems – MTNS 2010 • 5–9 July, 2010 • Budapest, Hungary.
- [21] “*A survey of model reduction by balanced truncation and some new results*”, A.C. Antoulas, S. Gugercin, INT. J. CONTROL, 20 MAY 2004, VOL. 77, NO. 8, 748–766.
- [22] “*Lectures on Dynamic Systems and Control, Chapter 26 - Balanced Realization*”, Mohammed Dahleh, Munther A. Dahleh, George Verghese, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.
- [23] “*Algorithms for model reduction of large dynamical systems*”, Thilo Penzl, Department of Mathematics and Statistics, University of Calgary, Calgary, Alta.
- [24] “*A Comparison of Balanced Truncation Techniques for Reduced Order Controllers*”, Katie A. E. Camp and Belinda B. King, Interdisciplinary Center for Applied Mathematics Virginia Tech.